# TIP_GIN_IA, TIP Generic Inventory Management Interface Information Agreement

**TIP_GIN_IA**

**Version 1.2**

October 2012

# Notice

# 1. Introduction

The TM Forum has given much attention to the evolution of next generation networks and the business and operational support systems needed to manage them. As a result, the Frameworx was developed to provide a toolkit of industry-agreed specifications and guidelines that cover key business and technical areas. This is reinforced by a set of unified open interfaces used between Operations Systems (OSs) for the purpose of network and service management. Many of these interfaces have been developed by individual groups such as OSS/J, mTOP and SLA management teams and may not be in alignment.

Considering the richness and complexity of the TM Forum Process Framework (eTOM), we may infer that Interfaces can only be complex and very specific to every problem space. While this is certainly true to some point, a deeper analysis reveals commonalities between problem areas that at first sight may look very different.

The Inventory team has deliberately taken the approach to identify capabilities that are frequently needed in management interfaces in various domains and to elaborate corresponding interface specifications supporting these capabilities using a domain neutral specification style. This approach leads to the availability of generic interfaces which set of capabilities are relevant in different context. It is especially true for task centric interfaces. These generic interfaces bring solutions beyond the Telecommunication business and can also be applicable to other domains, in line with the objectives of the Strategic Imperatives as outlined in the TM Forum Strategic plan.

Looking at the problem statement in the context of Telecommunication business only, the need for a flexible approach was expressed in a draft version of the “NGCOR Consolidated Requirements” elaborated by the NGMN Alliance (dated 18-July-2011) and submitted to the TM Forum as part the a liaison agreement. Section 6 of the "NGCOR Consolidated Requirements" is entirely devoted to “High Level OSS Requirements for Inventory Management”. R1 and R5 explicitly ask for the capability for the same Inventory Interface to manage different domains and areas for converged fixed-mobile environment: “R1: Capability to manage resource models of variety of technology infrastructure domains and areas of converged fixed-mobile environment.” “R5: Capability to manage service models of different domains and areas for converged fixed-mobile services.”

Four generic task centric service interfaces have been identified in a first round: Generic Query Interface, Generic Update Interface, Generic Notify Interface and Import-Export Interface. At the time of writing, only the Generic Query Interface has been specified in details; the other ones are under study.

This document presents the Generic Query Service Interface and the Generic Update Service Interface.

## 1.1. Document Structure

The following sections are contained in this document:

- Section 1 is the document introduction
- Section 2 provides a summary description of the interface
- Section 3 describes the Information Model used by this interface
- Section 4 describes all the Service Interfaces contained in this interface (Generic Query and Generic Update for the time being )

## 1.2. Conventions Used In This Document

In this document, we use the following color conventions.

In the attribute tables:

- An attribute shown on white background is local.
- An attribute shown on lavender background is inherited.

- An attribute shown on green background is implicit.

Similarly, text in green color indicates implicit data.

Implicit information does not appear in the model, but will be added by the generators in the final interface specification. Implicit data is defined in the Internal Framework Model. Please refer to the Framework Guidebook for details.

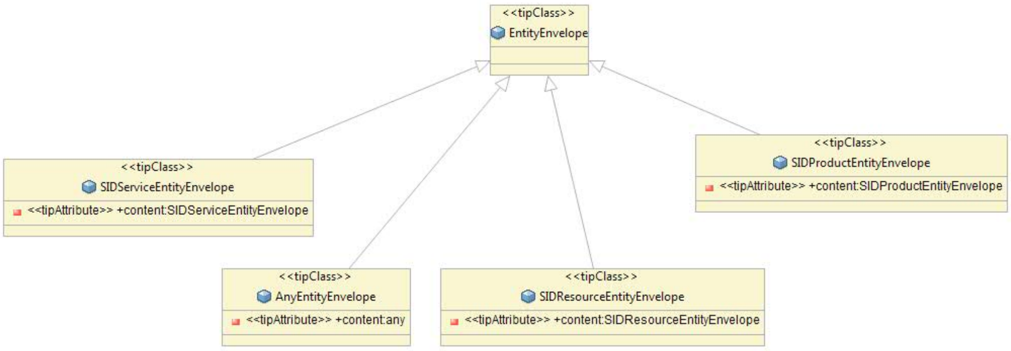
All links are with blue color.

# 2. Interface Summary

## 2.1. Information Model

This section presents the new artifacts that will used as input or output parameters of the various operations of the Inventory Interface, in addition to artifacts already available in the JOSIF framework.

The Inventory Interface is information model neutral. The adaptation of a generic interface to a given domain requests that these artifacts are "specialized" to support the corresponding information specific aspects. The number of Meta Classes artifacts introduced is very limited; the most important one is the concept of Entity Envelope.
An Envelope is made of four elements, one of them, the "Content", being used to encapsulate domain specific information of any nature, specified in another standard or as proprietary information. The encapsulated "Content" can be interpreted given the "Context" element available in the Envelope. In addition, the Object Class ("Type") and the "Identifier" of the encapsulated Managed Object are available in the Envelope without requesting the interpretation of the "Content" element. The following figure illustrate example of such Entity Envelopes where the "Content" may represent an artifact from any Information Models (e.g. SID, DMTF, Proprietary).


<<tipClass>>
EntityEnvelope
<<tipClass>>
SIDServiceEntityEnvelope
<<tipAttribute>> +content:SIDServiceEntityEnvelope
<<tipClass>>
SIDProductEntityEnvelope
<<tipAttribute>> +content:SIDProductEntityEnvelope
<<tipClass>>
AnyEntityEnvelope
<<tipAttribute>> +content:any
<<tipClass>>
SIDResourceEntityEnvelope
<<tipAttribute>> +content:SIDResourceEntityEnvelope

### 2.1.1. Entity Envelope

An Entity Envelope is an abstract model independent artifact used to encapsulate a model specific artifact. More precisely it is constituted of:

- An Entity Identifier element, taken from JOSIF
- A model specific artifact that represents any kind of managed entity. For example in the case the context is the TMF Information Model (a.k.a. SID) it may represent a SID Entity, a SID Entity Specification.

The Entity Identifier element contains three components:

- The context to interpret the model specific artifact encapsulated in the content component
- the unique identifier of the specific artifact
- the type of the specific artifact

The EntityEnvelope is a transient object used to encapsulate the content. The EntityEnvelope ID is the same as the encapsulated Managed Object ID.

### 2.1.2. Entity Envelopes Collection

An Entity Envelopes Collection is an unordered container of Entity Envelope items. Note 1: It may happen that the model specific artifacts contained in the Entity Envelopes elements are interrelated (with associations). It is only by looking at the details of the two model specific entities, that one can find this relation, since such association information is not explicitly conveyed by the Collection, as opposed to the Entity Envelopes Graph.

### 2.1.3. Association

An Association is an artifact used to represent a relationship between two Entity Envelope items. It is constituted of three components:

- The name of the association
- aEnd and zEnd, each representing the Entity Identifiers of the two unordered collection of Entity Envelopes related through the association.

Exemple: a given EQ (Equipment) relates to multiple PTPs (Physical Termination Point) through the “support” association. A single Association will be used, with the aEnd representing the Identifier of the EQ and the zEnd representing a collection of Identifiers for the supported PTPs.

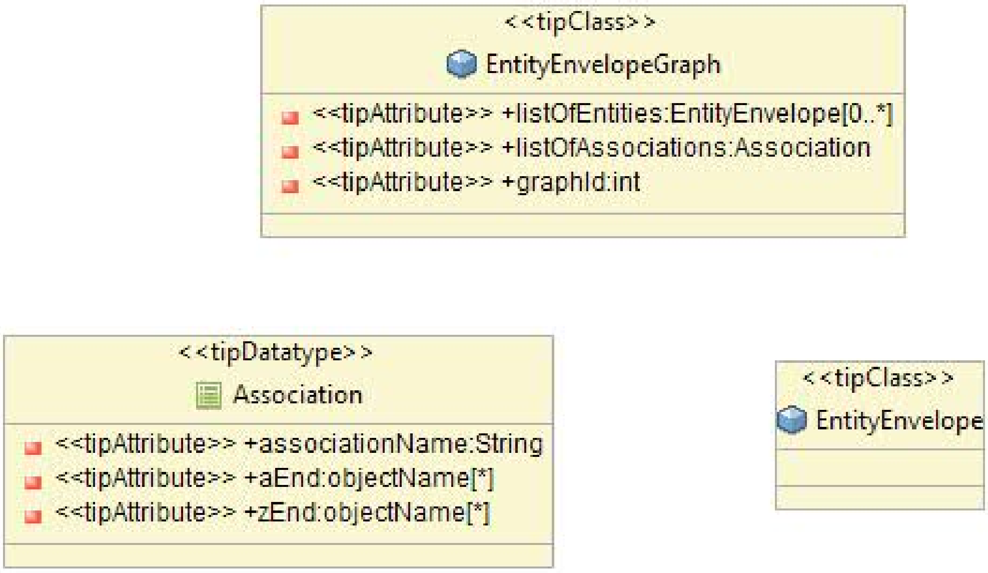
### 2.1.4. Entity Envelopes Graph

An Entity Envelope Graph is used to represent a set of Entity Envelopes and a set of Associations between them. More precisely it is constituted of two components:

- An Entity Envelopes Collection
- An Association Collection

For every aEnd or zEnd Entity Identifier in the Association Collection, the corresponding Entity Envelope must be present in the Entity Envelopes Collection. A Entity Envelopes Graph may represent a connected or a disconnected graph.

The graph object is shown in the figure below:

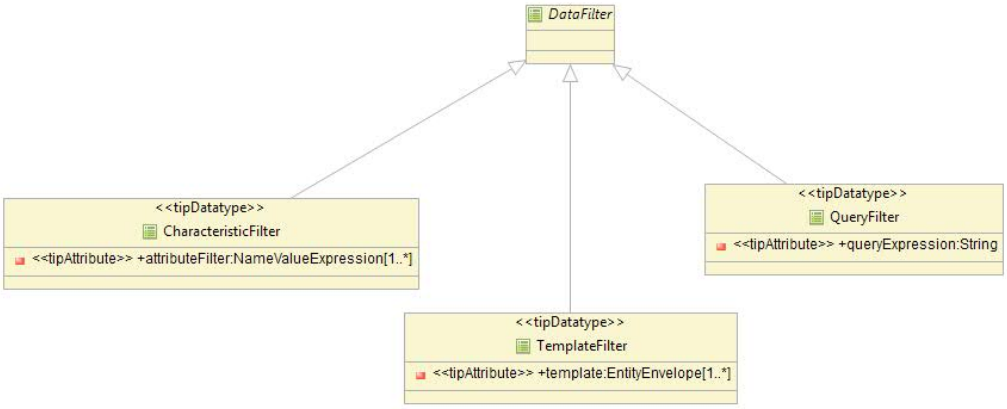

<<tipClass>>
EntityEnvelopeGraph
<<tipAttribute>> +listOfEntities:EntityEnvelope[0..*]
<<tipAttribute>> +listOfAssociations:Association
<<tipAttribute>> +graphId:int
<<tipDatatype>>
Association
<<tipAttribute>> +associationName:String
<<tipAttribute>> +aEnd:objectName[*]
<<tipAttribute>> +zEnd:objectName[*]
<<tipClass>>
EntityEnvelope

## 2.1.5. Filter

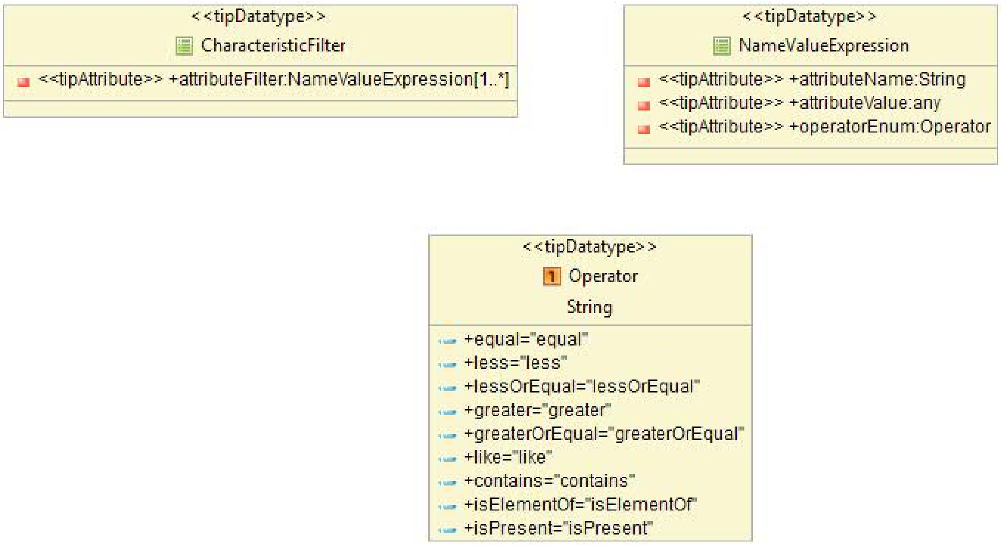
Three types of filters are used:

- Template Filter: for associative lookup
- Characteristic Filter: a logical expression where atomic terms represent valued attributes;
- Query Filter: for XPATH based filtering

The filters are shown in the figure below:


DataFilter
<<tipDatatype>>
CharacteristicFilter
<<tipAttribute>> +attributeFilter:NameValueExpression[1..*]
<<tipDatatype>>
QueryFilter
<<tipAttribute>> +queryExpression:String
<<tipDatatype>>
TemplateFilter
<<tipAttribute>> +template:EntityEnvelope[1..*]

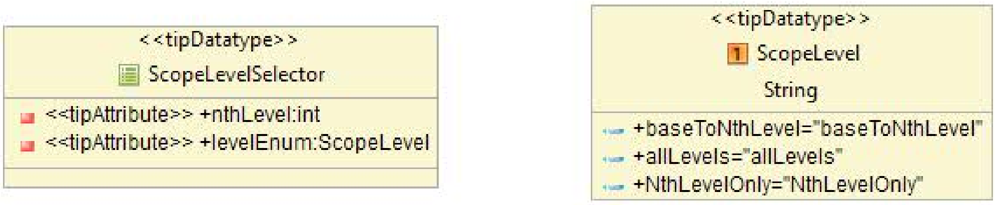
The characteristic filter is shown in the figure below:


<<tipDatatype>>
CharacteristicFilter
<<tipAttribute>> +attributeFilter:NameValueExpression[1..*]
<<tipDatatype>>
NameValueExpression
<<tipAttribute>> +attributeName:String
<<tipAttribute>> +attributeValue:any
<<tipAttribute>> +operatorEnum:Operator
<<tipDatatype>>
Operator
String
+equal="equal"
+less="less"
+lessOrEqual="lessOrEqual"
+greater="greater"
+greaterOrEqual="greaterOrEqual"
+like="like"
+contains="contains"
+isElementOf="isElementOf"
+isPresent="isPresent"

## 2.1.6. Scope

Scoping is used to target objects in containment trees.

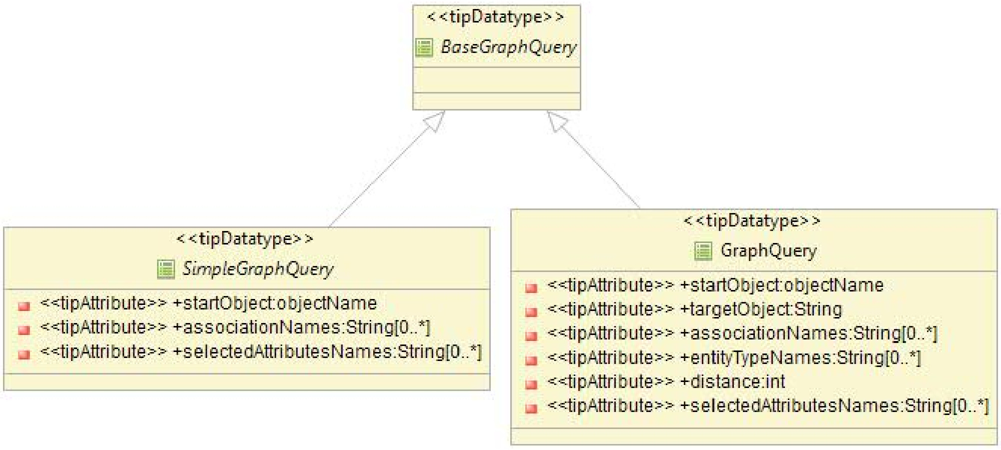
The scope objects are shown in the figure below:


<<tipDatatype>>
ScopeLevelSelector
<<tipAttribute>> +nthLevel:int
<<tipAttribute>> +levelEnum:ScopeLevel
<<tipDatatype>>
ScopeLevel
String
+baseToNthLevel="baseToNthLevel"
+allLevels="allLevels"
+NthLevelOnly="NthLevelOnly"

## 2.1.7. Graph Query

Graph Query objects are used for extracting a set of entities and associations given a combination of a starting entity, association names and target entity.

The Graph Query objects are shown in the figure below:


<<tipDatatype>>
BaseGraphQuery
<<tipDatatype>>
SimpleGraphQuery
<<tipAttribute>> +startObject:objectName
<<tipAttribute>> +associationNames:String[0..*]
<<tipAttribute>> +selectedAttributesNames:String[0..*]
<<tipDatatype>>
GraphQuery
<<tipAttribute>> +startObject:objectName
<<tipAttribute>> +targetObject:String
<<tipAttribute>> +associationNames:String[0..*]
<<tipAttribute>> +entityTypeNames:String[0..*]
<<tipAttribute>> +distance:int
<<tipAttribute>> +selectedAttributesNames:String[0..*]

### 2.1.8. Policies

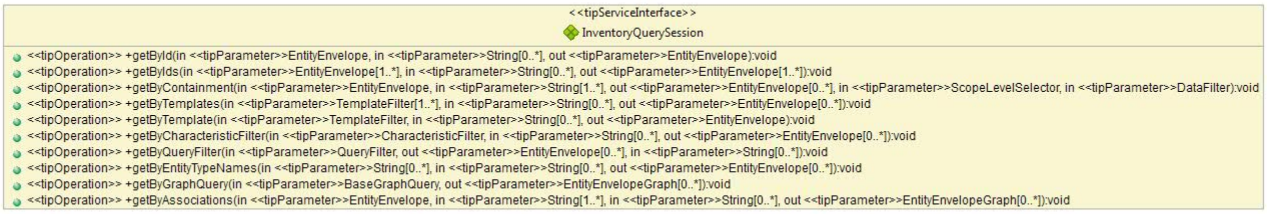
Sometimes the behavior of some dynamic operations depends on Policies (or Capabilities) supported by the Target System. A Policy has a name and a list of possible valid values. Generally a Policy is configured with a unique value, but it may happen that Policies are configured to support multiple values. The following Policies have been defined (more policies may be defined in the future):

- ModelConformanceValidationPolicy.This Policy indicates how the Target System validates conformance of information changes requested by any Requesting System against the Information Model it supports. Note that this standard is not prescriptive on the exact meaning of conformance; it is considered as an implementation matter. However, for any request, attribute values must always fit the constraints of the corresponding attribute type otherwise the request will raise an exception.
- MandatoryAssociationPolicy
- AttributeValuePolicy
- NamingPolicy. This policy applies to operations related to the creation of a new Object.

## 2.2. Query Session

The Query Session interface support the query of Entities based on a number of criteria

The Query Session Interface is shown in the figure below:



```
<<tipServiceInterface>>
InventoryQuerySession
<<tipOperation>> +getById(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[1..*]):void
<<tipOperation>> +getByContainment(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>ScopeLevelSelector, in <<tipParameter>>DataFilter):void
<<tipOperation>> +getByTemplates(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByTemplate(in <<tipParameter>>TemplateFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByCharacteristicFilter(in <<tipParameter>>CharacteristicFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByQueryFilter(in <<tipParameter>>QueryFilter, out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>String[0..*]):void
<<tipOperation>> +getByEntityTypeNames(in <<tipParameter>>String[0..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByGraphQuery(in <<tipParameter>>BaseGraphQuery, out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
<<tipOperation>> +getByAssociations(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
```

This section presents the different capabilities of the Query Interface giving the details of their semantics.

### 2.2.1Query a single Managed Object knowing its identifier

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to a Managed Object of interest as follows: The Requesting System specifies:

- an “Identifer” (mandatory): the identifier of the Managed Object of interest
- “Attribute Names” (mandatory): a list of Attribute Names that it owns; (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System returns an Entity Envelope containing the identified Managed Object with only the valued attributes corresponding to the list of Attribute Names.

### 2.2.2 Query several Managed Objects knowing their Identifiers

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Object of interest as follows. The Requesting System specifies:

- a list of “Identifiers” (mandatory): the identifiers of the Managed Objects of interest
- “Attribute Names” (mandatory): a list of Attribute Names that it owns; (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System return an Entity Envelopes Collection containing the identified Managed Objects with only the valued attributes corresponding to the list of Attribute Names.

Note 1: not all the Attribute Names may be present is every Object Class of the different Managed Objects identified.

Note 2: in case an identified Managed Objet does not contain any attribute from the Attribute Names list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

### 2.2.3 Query several Managed Objects in the same containment tree

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows. The Requesting System specifies:

- “Root Subtree” (mandatory): the identifier of a single Managed Object (the root of the subtree)
- “Level” (mandatory): a parameter indicating how many levels, in the containment tree, to select from the root Managed Object;
- A filter (optional): this filter may be a Query Filter or a Characteristics Filter; the Target System uses it to select from the Managed Objects already scoped in the containment tree only the ones matching the filter (called the Managed Objects of interest).
- “Attribute Names” (mandatory): a list of Attribute Names that it owns; (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System returns an Entity Envelopes Collection containing the Managed Objects of interest with only the valued attributes corresponding to the list of Attribute Names.

Note 1: querying several Managed Objects in the same containment tree is useful in the case where Managed Objects are organized in a containment tree structure and scoping in this containment tree is desired.

Note 2: not all the Attribute Names may be present in every Object Class of the different Managed Objects of interest.

Note 3: in case a Managed Objet in the scope does not contain any attribute from the Attribute Names list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

## 2.2.4 Query several Managed Objects matching the same Template

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies:

- A "Template" (mandatory): a list of valued attributes all belonging to a valid Object Class of the Information Model supported by the Target System; the Target System uses this list to identify all the Managed Objects which valued attributes match the supplied list (called the Managed Objects of interest).
- "Attribute Names" (mandatory): a list of Attribute Names that it owns; (an empty list means that all attributes are of interest; a list with the reserved name "EID" means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

For every Managed Object matching the Template, the Target System collects:

- the valued attributes corresponding to the list of Attribute Names
- and optionally the valued attribute already present in the Template

and encapsulates them as Managed Objects of the same Object Class as the Template; in turn the corresponding Managed Objects are encapsulated in Entity Envelopes inside an Entity Envelopes Collection which is returned.

Note 1: while a Template is built from a specific Object Class, it does not necessarily correspond to a valid Managed Object of this class; for example mandatory attributes of the Object Class may be omitted in the Template.

Note 2: the name of the Object Class from which the Template is built is an inherent part of the Template; for instance if class C1 and class C2 both contain the attribute "OperationalState" and other distinct attributes, a Template built from class C1 and containing "OperationalState" as a unique Attribute Name will differ from a Template built from class C2 also containing "OperationalState" as a unique Attribute Name.

Note 3: a degenerate form of this requirement is when the Template does not contain any valued attribute at all; in which case the only relevant information conveyed by the Template is the name of the Object Class of the Managed Objects of interest (see R_TMF_GQ_II_0012)

## 2.2.5 Query several Managed Objects matching the same Template

Query several Managed Objects matching any Template from a list The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies:

- "Templates" (mandatory): One or several independent lists of valued attributes, each list belonging to a valid Object Class of the Information Model supported by the Target System (called the templates); the Target System uses these lists to identify all the Managed Objects which valued attributes match at least one template (called the Managed Objects of interest).

- “Attribute Names” (mandatory): a list of Attribute Names from the Managed Object class of the different templates; (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

For every Managed Object matching at least one template, the Target System collects:

- the valued attributes corresponding to the list of Attribute Names
- and the valued attributes of the matched template (mandatory)

and encapsulates them as Managed Objects of the same Object Class as the matched template; in turn the corresponding Managed Objects are encapsulated in Entity Envelopes inside an Entity Envelopes Collection which is returned.

Note 1: while a Template is built from a specific Object Class, it does not necessarily correspond to a valid Managed Object of this class; for example mandatory attributes of the Object Class may be omitted in the Template.

Note 2: the name of the Object Class from which the Template is built is an inherent part of the Template; for instance if class C1 and class C2 both contain the attribute “OperationalState” and other distinct attributes, a Template built from class C1 and containing “OperationalState” as a unique attribute name will differ from a Template built from class C2 also containing “OperationalState” as a unique attribute name.

Note 3: a degenerate form of this requirement is when a Template in the list does not contain any valued attribute at all; in which case the only relevant information conveyed by such a Template is the name of the Object Class of the Managed Objects of interest (see R_TMF_GQ_II_0012).

Note 4: not all the Attribute Names may be present in every Object Class of the different Templates.

As example, if class C1 contains attributes AName1 and AName2 and if class C2 contains attributes AName1 and AName3 it is a valid request to send two Templates: one built from class C1 with valued attribute AName1 and one built from class C2 with valued attribute AName1 also (possibly with a different value) and a list of attribute names containing AName2 and AName3.

Note 5: in case a Managed Object of interest does not contain any attribute from the Attribute Name list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

### 2.2.6 Query several Managed Objects matching a Characteristics Filter

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies”

- a “Characteristics Filter” (mandatory): a logical expression where atomic terms represent valued attributes; this logical expression is called a characteristic filter
- “Attribute Names” (mandatory): a list of Attribute Names that the Managed Objects of interest may own (mandatory); (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System collects the Managed Objects filtered in by the Characteristics Filter with only the valued attributes corresponding to the list of Attribute Names; in turn the corresponding Managed Objects are encapsulated in Entity Envelopes inside an Entity Envelopes Collection which is returned.

Note 1: not all the attribute names may be present is every Object Class of the different Managed Objects filtered in.

Note 2: in case a filtered in Managed Objet does not contain any attribute from the attribute names list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

### 2.2.7Query several Managed Objects matching a Query Filter

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies”

- a “Query Filter” (mandatory)” an XPATH expression where the atomic terms represent either an attribute name or an attribute value
- “Attribute Names” (mandatory): a list of Attribute Names that the Managed Objects of interest may own (mandatory); (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System collects the Managed Objects filtered in by the Query Filter with only the valued attributes corresponding to the list of Attribute Names; in turn the corresponding Managed Objects are encapsulated in Entity Envelopes inside an Entity Envelopes Collection which is returned.

Note 1: not all the Attribute Names may be present is every Object Class of the different Managed Objects filtered in.

Note 2: in case a filtered in Managed Objet does not contain any attribute from the Attribute Names list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

### 2.2.8Query by Entity Type Names

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies:

- a list of “Entity Type Names” (mandatory): they represent names of Object Classes of the Information Model supported by the Target System; the Target System uses this list to identify all the Managed Objects from these Object Classes (called the Managed Objects of interest).
- “Attribute Names” (mandatory): a list of Attribute Names that the Managed Objects of interest may own (mandatory); (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

For every Managed Object for which the corresponding Object Class name is in the supplied list, the Target System collects the valued attributes corresponding to the list of Attribute Names and encapsulates them as Managed Objects of the corresponding Object Classes; in turn the corresponding Managed Objects are encapsulated in Entity Envelopes inside an Entity Envelopes Collection which is returned.

Note 1: not all the Attribute Names may be present is every named Object Class.

Note 2: in case a Managed Object of interest does not contain any attribute from the Attribute Names list, then the corresponding returned Entity Envelope will contain an empty Managed Object from the associated Object Class.

### 2.2.9Query a Graph of Managed Objects

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows: The Requesting System specifies:

- “Origin Object” (mandatory): the identifier of a Managed Object
- (exactly one of the three parameters below must be supplied) - “Target Object”: a Managed Object different from the “Origin Object” or - “Target Type”: an Object Class Name or - “Distance”: an Integer value
- “Association Names” (optional) a list of Association Names
- “Entity Type Names” (optional) a list of Object Class Names of the Information Model supported by the Target System
- “Attribute Names” (mandatory) a list of attribute names that the Managed Objects of interest may own; (an empty list means that all attributes are of interest; a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System selects the Managed Objects as follows:

- In a first step, the Target System selects all the Managed Objects that are navigable along any path from the Origin Object: - to the Target Object or - to any Managed Objects which Object Class Name corresponds to the Target Type or - that can be reached by a maximum of Distance vertices
- In a second step, the Target System may discard some elements from this ensemble of Managed Objects: - If Association Names is supplied, only the Managed Objects that can be navigated using one or several such associations are retained - If Entity Type Names is supplied, only the Managed Objects which Object Class Name is present in Entity Type Names are retained;

For every Managed Object in this final ensemble, the Target System retains only the valued attributes corresponding to the list of supplied Attribute Names, and encapsulates the corresponding Managed Objects in Entity Envelopes; complementary it creates a list of Associations possibly existing between the selected Managed Objects; all Associations between the selected Managed Objects must be collected (refer to R_TMF_GQ_I_0003). The two collections are returned together as separate items of an Entity Envelopes Graph (refer to R_TMF_GQ_I_0004).

### 2.2.10Query by Associations

The Interface supports the capability for a Requesting System to query the Target System in order to return valued attributes associated to multiple Managed Objects of interest as follows:

The Requesting System specifies:

- Origin Object (mandatory):the identifier of a Managed Object
- Association Names (mandatory) a list of Association Names
- Attribute Names(mandatory) a list of attribute names that the Managed Objects of interest may own; (an empty list means that all attributes are of interest; a list with the reserved name EID means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

The Target System selects the Managed Objects which can be navigated from the Origin Object using one or several associations which names are present in the Association Names.

For every Managed Object in this final ensemble, the Target System:

- retains only the valued attributes corresponding to the list of supplied Attribute Names, and encapsulates the corresponding Managed Objects in Entity Envelopes;

- it creates a list of Associations possibly existing between the selected Managed Objects; all direct Associations between the selected Managed Objects must be collected.

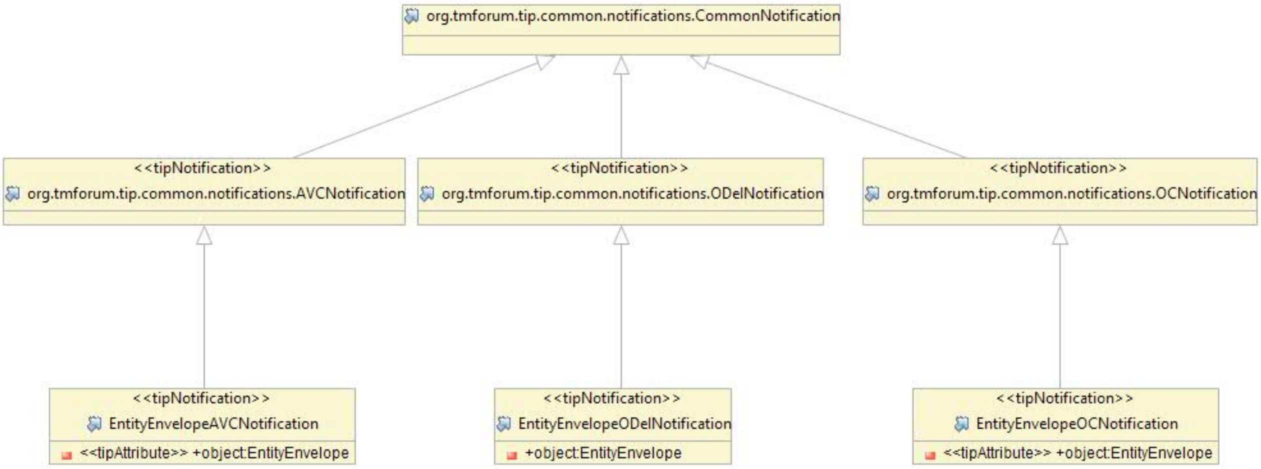
The two collections are returned together as separate items of an Entity Envelopes Graph .

### 2.2.11Notifications

This interface supports the generation of the following notifications:

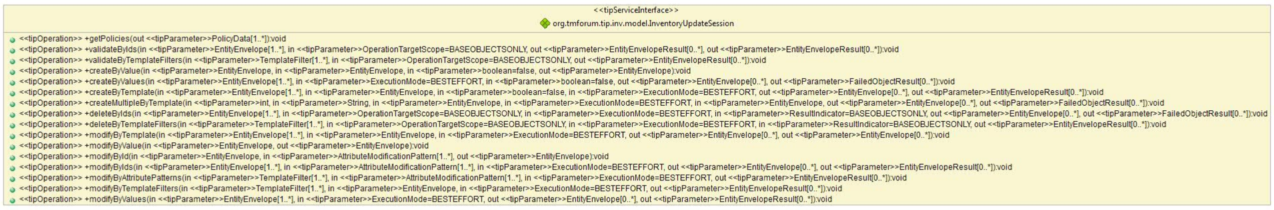
- Entity Envelope Object Create , Delete and Attribute Value Changes indicating Creation, Modification and Deletion of Entities
- HeartbeatNotification, used for detecting loss of communication between client and server

The notifications associated with the Inventory Management interface are shown in the figure below:


org.tmforum.tip.common.notifications.CommonNotification
<<tipNotification>>
org.tmforum.tip.common.notifications.AVCNotification
<<tipNotification>>
org.tmforum.tip.common.notifications.ODelNotification
<<tipNotification>>
org.tmforum.tip.common.notifications.OCNotification
<<tipNotification>>
EntityEnvelopeAVCNotification
<<tipAttribute>> +object:EntityEnvelope
<<tipNotification>>
EntityEnvelopeODelNotification
+object:EntityEnvelope
<<tipNotification>>
EntityEnvelopeOCNotification
<<tipAttribute>> +object:EntityEnvelope

## 2.3. Update Session

The Update Session Interface is shown in the figure below:



```
<<tipServiceInterface>>
org.tmforum.tip.inv.model.InventoryUpdateSession

<<tipOperation>> +getPolicies(out <<tipParameter>>PolicyData[1..*]):void
<<tipOperation>> +validateByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +validateByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +createByValue(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>EntityEnvelope, in <<tipParameter>>boolean=false, out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +createByValues(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>boolean=false, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +createByTemplate(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>boolean=false, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +createMultipleByTemplate(in <<tipParameter>>int, in <<tipParameter>>String, in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>EntityEnvelope, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +deleteByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>ResultIndicator=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +deleteByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>ResultIndicator=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByTemplate(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +modifyByValue(in <<tipParameter>>EntityEnvelope, out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +modifyById(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>AttributeModificationPattern[1..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +modifyByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>AttributeModificationPattern[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByAttributePatterns(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>AttributeModificationPattern[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByValues(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
```

This section presents the different capabilities of the Update Interface giving the details of their semantics.

### 2.3.1Get Policies

The Interface supports the capability for a Requesting System to query the Target System about which all policies related to the Update features

- There is no input parameter to be sent by the Target system.
- The Target System returns a list of all Policy Data. A Policy Data is a data structure made of the name of the Policy and the corresponding value(s) supported.

Note 1: this Interface does not expose any operation to set or modify the values of the policies supported by the Target system. It is assumed that these policies values are set at configuration time or at run time through another Interface or another mechanism.

## 2.3.2Validate Multiple Objects knowing their Identifiers

The Interface supports the capability for a Requesting System to request that the Target System check whether a set of Objects owned by the Target System and within the Scope of a one or more Base Objects known by their Identifiers conform or not to the constraints of the Information Model.

The Requesting System specifies:

- BaseObjectsToValidate (mandatory). A list of Entity Identifiers; this parameter represents the Identifiers of the Objects (Base Objects) to check for validation
- Scope (optional; default = BASEOBJECTSONLY): this parameter indicates whether dependent Objects must be implicitly validated or not.

The execution is considered successful if at least one Object could be checked for validation (this Object may be conformant or not).

The Target System returns:

- If no single Object from the BaseObjectsToValidate list could be checked for validation, then the execution has failed and an exception is returned

In any other case, the Target System returns:

- A list of all the Objects Identifiers which were checked for validation with, for each of them, an indication if the Object conformed or not, and if not an associated reason.
- A list (possibly empty) of Objects Identifiers which were NOT checked for validation and for each of them an associated reason (e.g. Object Not Found).

## 2.3.3Validate Multiple Objects exactly matching any TemplateFilter from a List

The Interface supports the capability for a Requesting System to request that the Target System check whether a set of Objects owned by the Target System and matching any given TemplateFilter from a list.

The Requesting System specifies:

- A list of TemplateFilter (mandatory): a Template is a set of attribute values all belonging to a single valid Object Class of the Information Model supported by the Target System; for every TemplateFilter, the Target System uses this set to identify all the Managed Objects which attribute values match all the ones supplied in the set (called the Managed Objects of interest). These Managed Objects will represent the Objects (Base Objects) to check for validation.
- Scope (optional; default = BASEOBJECTSONLY): this parameter indicates whether dependent Objects must be implicitly deleted or not.

The Target System returns: The execution is considered successful if at least one Object could be checked for validation (this Object may be conformant or not).

- If no Objects at all matches any of the supplied TemplateFilters, then the execution has failed and an exception is returned.
- Otherwise the Target System returns a list of all the Objects Identifiers which were checked for validation with, for each of them, an indication if the Object conformed or not, and if not an associated reason.

### 2.3.4Create a Single Object

The Interface supports the capability for a Requesting System to request that the Target System creates a new Managed Object as follows

The Requesting System specifies:

- The ObjectToCreate Entity Envelope (mandatory).
- A TemplateObjectReference (optional): it is an Entity Identifier referring to an Object that must exist in the Target System; the corresponding Object must be of the same Object Class or super Class as the Class of the Objects to create; if the two above conditions are not respected an exception is returned.
- IsAutoNamingRequested (optional; default = FALSE): a Boolean parameter driving the generation of the name.

The Target System creates an Object as follows:

- For every attribute in the Object Class, it uses the values supplied in the ObjectToCreate parameter in priority; then, if there are other attributes in the corresponding Class for which no value is available, it looks if a value is supplied for this attribute in the Object identified by the TemplateObjectReference parameter when it exists; at last, if there are still other attributes in the corresponding Class for which no value is available (in the ObjectToCreate or in the Object identified by the TemplateObjectReference), it looks in the Information Model to see if a default value exists.
- The name/identifier of the Object to create depends on the Naming Policy(ies) supported by the Target System and on the value of the IsAutoNamingRequested parameter.

The Target System returns:

- If the execution is successful: a copy of the created Object encapsulated in an Entity Envelope.
- If the execution has failed: an exception.

Note 1: The Class definition (with its attributes) will be used as reference for ALL other attributes found neither in ObjectToCreate nor TemplateObjectReference.

Note 2: The active policies (e.g. MandatoryAssociationPolicy) will apply at the end of this creation process: e.g. if the information model indicates that a given attribute is mandatory and no value can be found and the MandatoryAssociationPolicy is True, then the creation fails.

### 2.3.5Create Multiple Objects of the Same Class by Value

The Interface supports the capability for a Requesting System to request that the Target System creates multiple new Objects as follows:

The Requesting System specifies:

- ObjectsToCreate (mandatory): a list of Entity Envelopes.
- A TemplateObjectValue (optional): it is an Entity Envelope .
- IsAutoNamingRequested (optional; default = FALSE): a Boolean parameter driving the generation of the name.
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System creates as many Objects as elements in the ObjectsToCreate list parameter as follows:

- It takes the elements in the ObjectsToCreate list parameter one by one; for every attribute (as per the Class definition) of each Object to create: it looks if a value is supplied for this attribute in the element of the list; then, if after the step above, there are other attributes in the corresponding Class for which no value is supplied in the element, it looks if a value is supplied for this attribute in the TemplateObjectValue parameter when it exists; at last, if there are still other attributes in the corresponding Class for which no value is available (in the element or in the ValueObject), it looks in the Information Model to see if a default value exists.
- The names/identifiers of the Objects to create depends on the Naming Policy(ies) supported by the Target System and on the value of the IsAutoNamingRequested parameter.

The Target System returns:

- If the execution is successful: a list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope. In the case of partial success, a warning indication may also be returned.
- If the execution has failed: an exception.

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be created, it is expected that of the Target System will continue trying to create the remaining Instances.

Note 2: There is one specific Content for each Object to create and repeated Content is put in the TemplateObjectValue; in other words, the attribute values of the TemplateObjectValue apply to every created Object, if they are not already present in the specific Content of corresponding Object.

### 2.3.6Create Multiple Objects of the Same Class by Reference

The Interface supports the capability for a Requesting System to request that the Target System creates multiple new Objects as follows:

The Requesting System specifies:

- ObjectsToCreate (mandatory): a list of Entity Envelopes.
- A TemplateObjectReference (optional): it is an Entity Identifier referring to an Object that must exist in the Target System; the corresponding Object must be of the same Object Class or super Class as the Class of the Objects to create; if the two above conditions are not respected an exception is returned.
- IsAutoNamingRequested (optional; default = FALSE): a Boolean parameter driving the generation of the name.
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System creates as many Objects as elements in the ObjectsToCreate list parameter as follows:

- It takes the elements in the ObjectsToCreate list parameter one by one; for every attribute (as per the Class definition) of each Object to create: it looks if a value is supplied for this attribute in the element of the list; then, if after the step above, there are other attributes in the corresponding Class for which no value is supplied in the element, it looks if a value is supplied for this attribute in the Object identified by the TemplateObjectReference parameter when it exists; at last, if there are still other attributes in the corresponding Class for which no value is available (in the element or in the Object identified by the TemplateObjectReference), it looks in the Information Model to see if a default value exists.

- The names/identifiers of the Objects to create depend on the Naming Policy(ies) supported by the Target System and on the value of the IsAutoNamingRequested parameter .

The Target System returns:

- If the execution is successful: a list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope. In the case of partial success, a warning indication may also be returned.
- If the execution has failed: an exception.

Note 1: If the execution mode is Best Effort, and if one of the Object could not be created, it is expected that the Target System will continue trying to create the remaining Instances.

Note 2: There is one specific Content for each Object to create and repeated Content is put in the TemplateObjectReference; in other words, the attribute values of the TemplateObjectReference apply to every created Object, if they are not already present in the specific Content of corresponding Object.

### 2.3.7Create n Objects of the Same Class by Value

In the case when the Requesting System delegates to the Target System the responsibility to construct the names of the Objects to create, this operation may be used; Applicable only if: NamingPolicy == ServerControl NamingPolicy == Both The Interface supports the capability for a Requesting System to request that the Target System create multiple new Objects as follows:

The Requesting System specifies:

- NumberOfObjectsToCreate (mandatory): an integer value indicating how many Objects must be created.
- ObjectClassName (mandatory) The name of the Object Class of the Objects to create.
- A TemplateObjectValue (optional): it is an Entity Envelope.
- An Object Identifier (mandatory): it is an Entity Identifier referring, when not empty, to an Object that must exist in the Target System; the corresponding Object will be the superior of all the Objects to create (empty in the case the Objects to create are at the root level).
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System creates NumberOfObjectsToCreate Objects as follows:

- For each attribute (as per the Class definition) of each Object to create: it looks if a value is supplied for this attribute in the TemplateObjectValue parameter when it exists; then if there are still other attributes in the corresponding Class for which no value is supplied in the TemplateObjectValue, it looks in the Information Model to see if a default value exists.
- The names/identifiers of the Objects to create must be constructed by the Target System; if there is a Naming Scheme and the Class of the Objects to create indicates that the Objects to create are not top level Objects, then the Target System will use the Object Identifier parameter to create each Object as a subordinate (the value of the RDN is the responsibility of the Target System).

The Target System returns:

- If the execution is successful: a list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope. In the case of partial success, a warning indication may also be returned.
- If the execution has failed: an exception.

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be created, it is expected that of the Target System will continue trying to create the remaining Instances.

Note 2: There is no specific Content for each Object to create; repeated Content is put in the TemplateObjectValue and apply to every created Object.

### 2.3.8Create n Objects of the Same Class By Reference

In the case when the Requesting System delegates to the Target System the responsibility to construct the names of the Objects to create, this operation may be used; Applicable only if: NamingPolicy == ServerControl NamingPolicy == Both The Interface supports the capability for a Requesting System to request that the Target System create multiple new Objects as follows:

The Requesting System specifies:

- NumberOfObjectsToCreate (mandatory): an integer value indicating how many Objects must be created.
- ObjectClassName (mandatory): the name of the Object Class of the Objects to create.
- A TemplateObjectReference (optional): it is an Entity Identifier referring to an Object that must exist in the Target System; the corresponding Object must be of the same Object Class or super Class as the Class of the Objects to create; if the two above conditions are not respected an exception is returned.
- An Object Identifier (mandatory): it is an Entity Identifier referring, when not empty, to an Object that must exist in the Target System; the corresponding Object will be the superior of all the Objects to create (empty in the case the Objects to create are at the root level).
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System creates NumberOfObjectsToCreate Objects as follows:

- For each attribute (as per the Class definition) of each Object to create: it looks if a value is supplied for this attribute in the Object identified by the TemplateObjectReference parameter when it exists; then if there are still other attributes in the corresponding Class for which no value is supplied in the ReferenceObject, it looks in the Information Model to see if a default value exists.
- The names/identifiers of the Objects to create must be constructed by the Target System; if there is a Naming Scheme and the Class of the Objects to create indicates that the Objects to create are not top level Objects, then the Target System will use the Object Identifier parameter to create each Object as a subordinate (the value of the RDN is the responsibility of the Target System).

The Target System returns:

- If the execution is successful: a list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope. In the case of partial success, a warning indication may also be returned.
- If the execution has failed: an exception.

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be created, it is expected that of the Target System will continue trying to create the remaining Instances.

Note 2: There is no specific Content for each Object to create; repeated Content is put in the TemplateObjectReference and apply to every created Object.

## 2.3.9Create Multiple Objects of Multiple Classes

The Interface supports the capability for a Requesting System to request that the Target System creates multiple new Objects as follows:

The Requesting System specifies:

- ObjectsToCreate (mandatory): a list of Entity Envelopes.
- IsAutoNamingRequested (optional; default = FALSE): a Boolean parameter driving the generation of the name.
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System creates as many Objects as elements in the ObjectsToCreate list parameter as follows:

- It takes the elements in the ObjectsToCreate list parameter one by one; for every attribute (as per the corresponding Class definition) of each Object to create: it looks if a value is supplied for this attribute in the element of the list; then, if there are still other attributes in the corresponding Class for which no value is available (in the element), it looks in the Information Model to see if a default value exists.
- The name/identifier of the Objects to create depend on the Naming Policy(ies) supported by the Target System and on the value of the IsAutoNamingRequested parameter.

The Target System returns:

- If the execution is successful: a list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope. In the case of partial success, a warning indication may also be returned.
- If the execution has failed: an exception.

Note 1: If the execution mode is Best Effort, and if one of the Object could not be created, it is expected that the Target System will continue trying to create the remaining Instances.

Note 2: This operation may be used to create a graph with cycles: an association between two Objects is represented by an attribute which value type is an Identifier; even in the case the MandatoryAssociationPolicy supported by the Target System is Strict, it is assumed that it is only at the end of the creation execution (once all the Objects to create have been created) that the validity of the associations are verified; e.g. an Equipment and a PTP showing a bidirectional association (supporting / supportedBy) can be created using a single request of this operation.

## 2.3.10Modify a Single Object knowing its Identifier

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- The ModifiedObject Entity Envelope (mandatory)

The Target System modifies the Object as follows: For every attribute in the ModifiedObject, the Target system modifies the corresponding attribute in the Object to modify (other attributes, not present in the ModifiedObject, are unchanged).

The Target System returns:

- If the execution is successful: the modified Object encapsulated in an Entity Envelope.
- If the execution fails: an exception.

### 2.3.11Modify a Single Object with Modification Patterns knowing its Identifier

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- IdentifierOfObjectToModify (mandatory): the Identifier of the Object to modify; the corresponding Object must exist in the Target System, otherwise the operation fails and an exception is returned.
- A set of Attribute Modification Patterns (mandatory); An Attribute Modification Patterns contains the following information:

- An Attribute name (mandatory)
- One or several Attribute Value(s): the value(s) to be used in the modification of the Attribute.The usage of this parameter is defined by the Modify operator. This parameter is optional when the Modify operator is set to default.
- A Modify operator (optional; default = SET): this operator drives the way the attribute value(s) (if supplied) is(are) to be applied to the attribute

The Target System returns:

- If the execution is successful: the modified Object encapsulated in an Entity Envelope.
- If the execution fails: an exception

### 2.3.12Modify Multiple Objects of the Same Class by Value knowing their Identifiers

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- ModifiedObjects (mandatory): a list of Entity Envelopes.
- A TemplateObjectValue (optional).
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System modifies the Objects identified in the ModifiedObjects list parameter as follows: it takes the Objects identified in the ModifiedObjects list parameter one by one. For every attribute in the corresponding ModifiedObject parameter the Target system modifies the corresponding attribute; Then if there are other attributes in the TemplateObjectValue, their values are used to modify the Object as well. (other attributes, not present in the ModifiedObject or in the TemplateObjectValue, are unchanged).

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in an Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

### 2.3.13Modify Multiple Objects of the Same Class by Reference knowing their Identifiers

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- ModifiedObjects (mandatory): a list of Entity Envelopes.
- A TemplateObjectReference (optional): it is an Entity Identifier referring to an Object that must exist in the Target System; the corresponding Object must be of the same Object Class or super Class as the Class of the Objects to modify; if the two above conditions are not respected an exception is returned.
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System modifies the Objects identified in the ModifiedObjects list parameter as follows: it takes the Objects identified in the ModifiedObjects list parameter one by one. For every attribute in the corresponding ModifiedObject parameter the Target system modifies the corresponding attribute; Then if there are other attributes in the Object identified by the TemplateObjectReference, their values are used to modify the Object as well. (other attributes, not present in the ModifiedObject or in the TemplateObjectValue, are unchanged)

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in an Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

### 2.3.14Modify Multiple Objects of the Same Class by Value exactly matching any TemplateFilter from a List

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- A list of TemplateFilters (mandatory): a TemplateFilter contains set of attribute values all belonging to a single valid Object Class of the Information Model supported by the Target System; for every TemplateFilter, the Target System uses this set to identify all the Managed Objects which attribute values match all the ones supplied in the set. These Managed Objects will represent the Objects to modify. All TemplateFilters must be of the same class.

- A TemplateObjectValue (mandatory).
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System modifies the Objects to modify as follows: It processes the Objects to modify one by one. For every attribute in the TemplateObjectValue, the Target system modifies the corresponding attribute in any Object to modify (other attributes, not present in the TemplateObjectValue, are unchanged).

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in an Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

### 2.3.15Modify Multiple Objects of the Same Class by Reference exactly matching any TemplateFilter from a List

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- A list of TemplateFilters (mandatory): a TemplateFilter contains set of attribute values all belonging to a single valid Object Class of the Information Model supported by the Target System; for every TemplateFilter, the Target System uses this set to identify all the Managed Objects which attribute values match all the ones supplied in the set. These Managed Objects will represent the Objects to modify. All TemplateFilters must be of the same class.
- A TemplateObjectReference (mandatory) it is an Entity Identifier referring to an Object that must exist in the Target System; the corresponding Object must be of the same Object Class or super Class as the Class of the Objects to modify; if the two above conditions are not respected an exception is returned.
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System modifies the Objects to modify as follows: It processes the Objects to modify one by one. For every attribute in the Object identified by the TemplateObjectReference parameter, the Target system modifies the corresponding attribute in any Object to modify (other attributes, not present in the Object identified by the TemplateObjectReference, are unchanged).

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in an Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

## 2.3.16Modify Multiple Objects with Modification Patterns knowing their Identifiers

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- IdentifiersOfObjectsToModify (mandatory): a list of Identifiers (mandatory): this parameter represents the Identifiers of the Objects to modify; the corresponding Objects must exist in the Target System, otherwise the operation fails and an exception is returned.
- A set of Attribute Modification Patterns (mandatory); this same set will apply to every Object to modify; an Attribute Modification Patterns contains the following information:

- An Attribute name (mandatory)
- One or several Attribute Value(s): the value(s) to be used in the modification of the Attribute. The usage of this parameter is defined by the Modify operator. This parameter is optional when the Modify operator is set to default.
- A Modify operator (optional; default = SET): this operator drives the way the attribute value(s) (if supplied) is(are) to be applied to the attribute
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

## 2.3.17Modify Multiple Objects with Modification Patterns exactly matching any TemplateFilter from a List

The Interface supports the capability for a Requesting System to request that the Target System modify an existing Object as follows:

The Requesting System specifies:

- A list of TemplateFilters (mandatory): a TemplateFilter is a set of attribute values all belonging to a single valid Object Class of the Information Model supported by the Target System; for every TemplateFilter, the Target System uses this set to identify all the Managed Objects which attribute values match all the ones supplied in the set (called the Managed Objects of interest). These Managed Objects will represent the Objects (Base Objects) to modify.
- A set of Attribute Modification Patterns (mandatory); this same set will apply to every Object to modify; an Attribute Modification Patterns contains the following information:

- An Attribute name (mandatory)

- One or several Attribute Value(s): the value(s) to be used in the modification of the Attribute. The usage of this parameter is defined by the Modify operator. This parameter is optional when the Modify operator is set to default.
- A Modify operator (optional; default = SET): this operator drives the way the attribute value(s) (if supplied) is(are) to be applied to the attribute
- Execution Mode (optional; default = BestEffort): this parameter indicates whether the execution should be BestEffort or Atomic.

The Target System returns:

- If the execution is fully or partially successful: the modified Objects encapsulated in Entity Envelopes array.
- If the execution fails: an exception

Note 1: If the execution mode is Best Effort, and if one of the Objects could not be modified, it is expected that the Target System will continue trying to modify the remaining Objects.

### 2.3.18Delete Multiple Objects knowing their Identifiers

The Interface supports the capability for a Requesting System to request that the Target System delete multiple Managed Object as follows:

The Requesting System specifies:

- BaseObjectsToDelete (mandatory): a list of Entity Identifiers; this parameter represents the Identifiers of the Objects (Base Objects) to delete.
- Scope (optional; default = BASEOBJECTSONLY): this parameter indicates whether dependent Objects must be implicitly deleted or not.
- ExecutionMode (optional; default = BESTEFFORT).
- ResultIndicator (optional; default = BASEOBJECTSONLY).

The Target System returns:

- If the execution is fully or partially successful: the Identifiers of the deleted Objects, in conformance with the ResultIndicator); in addition, in case of partial success, the Target System may also return the list of the Base Objects Identifiers which could not be deleted with, for each of them, an associated reason.
- If the execution has failed: an exception, and the list of the Base Objects Identifiers which could not be deleted with, for each of them, an associated reason.

Note 1: If the execution mode is Best Effort, and if one of the Base Objects could not be deleted, it is expected that the Target System will continue trying to remove the remaining Base Objects.

Note 2: whatever Scope or Execution Mode value is supplied, the Target System must always execute this request in full respect of the constraints specified in the Information Model; as an example, consider the SNC (SubNetwork Connection) and TP (Termination Point) Object Classes. The SNC contains attributes such as aEndPoint, zEndPoint which values are references to TPs Object Instances. The Object Model may enforce that a SNC Object Instance must always have non null values for its aEndPoint, zEndPoint attributes. In this case it is not possible to remove TP Object Instances if there is still an SNC Object Instance referencing to them; to complete this example, it may also happen that the rules of the Information Model allow the existence of SNC Object Instances with null valued aEndPoint, zEndPoint attributes only at creation time, but prevent those attributes to be set again to null reference values at a later stage. In any case, the rules associated to the Information Model should be enforced during the execution of this operation.

### 2.3.19Delete Multiple Objects exactly matching any TemplateFilter from a List

The Interface supports the capability for a Requesting System to request that the Target System delete multiple Managed Object matching any given Template from a list.

The Requesting System specifies:

- A list of TemplateFilters (mandatory). For every TemplateFilter, the Target System uses this set to identify all the Managed Objects which attribute values match all the ones supplied in the set (called the Managed Objects of interest). These Managed Objects will represent the Objects (Base Objects) to delete.
- Scope (optional; default = BASEOBJECTSONLY): this parameter indicates whether dependent Objects must be implicitly deleted or not.
- ExecutionMode (optional; default = BESTEFFORT).
- ResultIndicator (optional; default = BASEOBJECTSONLY).

The Target System returns:

- If the execution is fully or partially successful: the Identifiers of the deleted Objects, in conformance with the ResultIndicator); in addition, in case of partial success, the Target System may also return the list of the Base Objects Identifiers which could not be deleted with, for each of them, an associated reason.
- If the execution has failed: an exception, and the list of the Base Objects Identifiers which could not be deleted with, for each of them, an associated reason.

Note 1: If the execution mode is Best Effort, and if one of the Base Objects could not be deleted, it is expected that the Target System will continue trying to remove the remaining Base Objects.

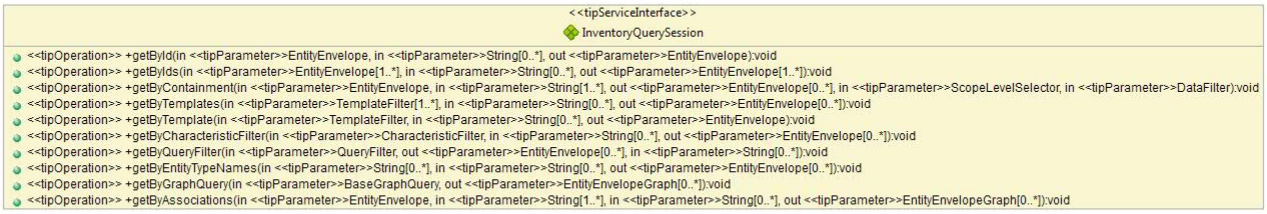
Note 2: whatever Scope or Execution Mode value is supplied, the Target System must always execute this request in full respect of the constraints specified in the Information Model; as an example, consider the SNC (SubNetwork Connection) and TP (Termination Point) Object Classes. The SNC contains attributes such as aEndPoint, zEndPoint which values are references to TPs Object Instances. The Object Model may enforce that a SNC Object Instance must always have non null values for its aEndPoint, zEndPoint attributes. In this case it is not possible to remove TP Object Instances if there is still an SNC Object Instance referencing to them; to complete this example, it may also happen that the rules of the Information Model allow the existence of SNC Object Instances with null valued aEndPoint, zEndPoint attributes only at creation time, but prevent those attributes to be set again to null reference values at a later stage. In any case, the rules associated to the Information Model should be enforced during the execution of this operation.

## 2.4. Interface Summary

The Inventory interfaces are information model neutral. The adaptation of a generic interface to a given domain requests that these artifacts are “specialized” to support the corresponding information specific aspects.

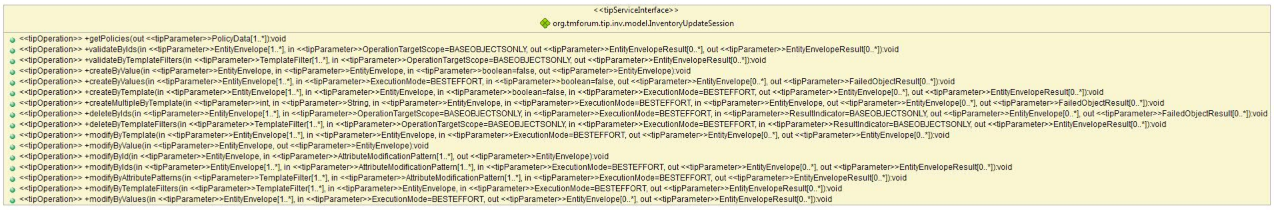
The Inventory interfaces provide polymorphic query and update operations relative to the Managed Entities. An Envelope mechanism is used to transfer the content of the Inventory System.

The picture below presents a view of the Query Session interface



```
<<tipServiceInterface>>
InventoryQuerySession
<<tipOperation>> +getById(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[1..*]):void
<<tipOperation>> +getByContainment(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>ScopeLevelSelector, in <<tipParameter>>DataFilter):void
<<tipOperation>> +getByTemplates(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByTemplate(in <<tipParameter>>TemplateFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByCharacteristicFilter(in <<tipParameter>>CharacteristicFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByQueryFilter(in <<tipParameter>>QueryFilter, out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>String[0..*]):void
<<tipOperation>> +getByEntityTypeNames(in <<tipParameter>>String[0..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByGraphQuery(in <<tipParameter>>BaseGraphQuery, out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
<<tipOperation>> +getByAssociations(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
```

The Update Session Interface is shown in the figure below:



```
<<tipServiceInterface>>
org.tmforum.tip.inv.model.InventoryUpdateSession
<<tipOperation>> +getPolicies(out <<tipParameter>>PolicyData[1..*]):void
<<tipOperation>> +validateByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +validateByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +createByValue(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>EntityEnvelope, in <<tipParameter>>boolean=false, out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +createByValues(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>boolean=false, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +createByTemplate(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>boolean=false, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +createMultipleByTemplate(in <<tipParameter>>int, in <<tipParameter>>String, in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>EntityEnvelope, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +deleteByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>ResultIndicator=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>FailedObjectResult[0..*]):void
<<tipOperation>> +deleteByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>OperationTargetScope=BASEOBJECTSONLY, in <<tipParameter>>ExecutionMode=BESTEFFORT, in <<tipParameter>>ResultIndicator=BASEOBJECTSONLY, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByTemplate(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +modifyByValue(in <<tipParameter>>EntityEnvelope, out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +modifyById(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>AttributeModificationPattern[1..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +modifyByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>AttributeModificationPattern[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByAttributePatterns(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>AttributeModificationPattern[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByTemplateFilters(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>EntityEnvelope, in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelopeResult[0..*]):void
<<tipOperation>> +modifyByValues(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>ExecutionMode=BESTEFFORT, out <<tipParameter>>EntityEnvelope[0..*], out <<tipParameter>>EntityEnvelopeResult[0..*]):void
```

# 3. Information Model

Packages available from TIP Generic Inventory Management Interface:
- org.tmforum.tip.inv.model

## 3.1. Package org.tmforum.tip.inv.model

## 3.1.1. Entities

### 3.1.1.1. AnyEntityEnvelope

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:
    org.tmforum.tip.inv.model.EntityEnvelope
    org.tmforum.tip.internal.entity.EntityBase
- Properties:
    This entity is mandatory
    This entity is extendable

**3.1.1.1.1. Attributes**

| name | datatype | properties | description |
|---|---|---|---|
| content | any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC disabled | |
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

**3.1.1.1.2. Associations**

There are no associations (local or inherited) available.

### 3.1.1.2. EntityEnvelope

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

org.tmforum.tip.internal.entity.EntityBase

- Description:

Entity Envelope artifactAn Entity Envelope is an abstract model independent artifact used to encapsulate a model specific artifact.More precisely it is constituted of:• An Entity Identifier (Ref [1]) element, taken from JOSIF, which three components indicating The context to interpret the model specific artifact encapsulated in the content component the unique identifier of the specific artifact the type of the specific artifact• A model specific artifact that represents any kind of managed entity.For example in the case the context is the TMF Information Model (a.k.a. SID) it may represent a SID Entity, a SID Entity Specification.

- Properties:

This entity is mandatory

This entity is extendable

This entity generates Object Creation notifications

This entity generates Object Deletion notifications

This entity does not generate Object Discovery notifications (NA)

### 3.1.1.2.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

### 3.1.1.2.2. Associations

There are no associations (local or inherited) available.

## 3.1.1.3. EntityEnvelopeGraph

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

org.tmforum.tip.internal.entity.EntityBase

- Description:

An EntityEnvelopeGraph is: a collection of EntityEnvelope and a collection of Associations where each association represents a relationship between specific Entities known through their id.

Entity Envelopes Graph artifactAn Entity Envelope Graph is used to represent a set of Entity Envelopes and a set of Associations between them.More precisely it is constituted of two components:• An Entity Envelopes Collection• An Association CollectionFor every aEnd or zEnd Entity Identifier in the Association Collection, the corresponding Entity Envelope must be present in the Entity Envelopes Collection.A Entity Envelopes Graph may represent a connected or a disconnected graph.

- Properties:

  This entity is optional

  This entity is extendable

### 3.1.1.3.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| listOfEntities | EntityEnvelope | - multiplicity is *<br>- read only<br>- unique<br>- passed by value<br>- optional<br>- AVC disabled | |
| listOfAssociations | Association | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC disabled | |
| graphId | int | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC disabled | |
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

### 3.1.1.3.2. Associations

There are no associations (local or inherited) available.

## 3.1.1.4. EntityEnvelopeResult

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

  org.tmforum.tip.inv.model.EntityEnvelope

  org.tmforum.tip.internal.entity.EntityBase

- Properties:

  This entity is optional

  This entity is extendable

### 3.1.1.4.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| failure | boolean | - multiplicity is 1<br>- unique<br>- mandatory<br>- AVC disabled | Indicate if the Entity is associated with a failure |
| reasonForFailure | String | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | |
| entityEnvelope | EntityEnvelope | - multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory<br>- AVC disabled | |
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

### 3.1.1.4.2. Associations

There are no associations (local or inherited) available.

## 3.1.1.5. SIDProductEntityEnvelope

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

  org.tmforum.tip.inv.model.EntityEnvelope

  org.tmforum.tip.internal.entity.EntityBase
- Properties:

  This entity is mandatory

  This entity is extendable

  This entity does not generate Object Creation notifications

  This entity does not generate Object Deletion notifications

  This entity does not generate Object Discovery notifications

### 3.1.1.5.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| content | Product | - multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory<br>- AVC disabled | |

| | | | |
|---|---|---|---|
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

### 3.1.1.5.2. Associations

There are no associations (local or inherited) available.

## 3.1.1.6. SIDResourceEntityEnvelope

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:
    - org.tmforum.tip.inv.model.EntityEnvelope
    - org.tmforum.tip.internal.entity.EntityBase
- Properties:
    - This entity is optional
    - This entity is extendable
    - This entity does not generate Object Creation notifications
    - This entity does not generate Object Deletion notifications
    - This entity does not generate Object Discovery notifications

### 3.1.1.6.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| content | Resource | - multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory<br>- AVC disabled | |
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

#### 3.1.1.6.2. Associations

There are no associations (local or inherited) available.

### 3.1.1.7. SIDServiceEntityEnvelope

- Type: Entity Artifact
- Package: org.tmforum.tip.inv.model
- All super types:
  - org.tmforum.tip.inv.model.EntityEnvelope
  - org.tmforum.tip.internal.entity.EntityBase
- Properties:
  - This entity is optional
  - This entity is extendable
  - This entity does not generate Object Creation notifications
  - This entity does not generate Object Deletion notifications
  - This entity does not generate Object Discovery notifications

#### 3.1.1.7.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| content | Service | - multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory<br>- AVC disabled | |
| identifer | EntityIdentifier | - multiplicity is 1<br>- unique<br>- invariant<br>- mandatory<br>- AVC enabled | The entity instance identifier EID. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |

#### 3.1.1.7.2. Associations

There are no associations (local or inherited) available.

## 3.1.2. Data Types

### 3.1.2.1. Association

- Type: Datatype Artifact

- Package: org.tmforum.tip.inv.model
- Description:

Association artifactAn Association is an artifact used to represent a relationship between two Entity Envelope items.It is constituted of three components:• The name of the association• aEnd and zEnd, each of type Entity Identifier representing the Entity Identifier of the two Entity Envelopes related through the association.

- Properties:

This datatype is extendable

### 3.1.2.1.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| associationName | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| aEnd | objectName | - multiplicity is *<br>- unique<br>- optional | |
| zEnd | objectName | - multiplicity is *<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.2. AttributeModificationPattern

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.2.2.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| name | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| value | any | - multiplicity is 0..1<br>- unique<br>- optional | |
| pattern | AttributeModificationPatternOperator | - multiplicity is 0..1<br>- unique<br>- optional | |

## 3.1.2.3. BaseGraphQuery

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Description:

A base type for all Graph Query types.

- Properties:

  This datatype is abstract

  This datatype is not extendable

#### 3.1.2.3.1. Attributes

There are no attributes (local or inherited) available.

### 3.1.2.4. CharacteristicFilter

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

  org.tmforum.tip.inv.model.DataFilter
- Description:

A logical expression where atomic terms represent valued attributes;this logical expression is called a characteristic filter

- Properties:

  This datatype is not extendable

#### 3.1.2.4.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| attributeFilter | NameValueExpression | - multiplicity is 0..*<br>- unique<br>- optional | |

### 3.1.2.5. DataFilter

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

  This datatype is abstract

#### 3.1.2.5.1. Attributes

There are no attributes (local or inherited) available.

### 3.1.2.6. FailedObjectResult

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

This datatype is not extendable

#### 3.1.2.6.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| failedObject | EntityEnvelope | - multiplicity is 1<br>- unique<br>- passed by id<br>- mandatory | |
| reasonForFailure | String | - multiplicity is 1<br>- unique<br>- mandatory | |

## 3.1.2.7. GraphQuery

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

org.tmforum.tip.inv.model.BaseGraphQuery

- Description:

“Origin Object” (mandatory):the identifier of a Managed Object (exactly one of the three parameters below must be supplied) “Target Object”: a Managed Object different from the “Origin Object” or “Target Type”: an Object Class Name or “Distance”: an Integer value “Association Names” (optional)a list of Association Names “Entity Type Names” (optional)a list of Object Class Names of the Information Model supported by the Target System “Attribute Names” (mandatory)a list of attribute names that the Managed Objects of interest may own;(an empty list means that all attributes are of interest;a list with the reserved name “EID” means that no attribute must be selected; to the exception of the Identifier of the corresponding Managed Object)

- Properties:

This datatype is extendable

#### 3.1.2.7.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| startObject | objectName | - multiplicity is 0..1<br>- unique<br>- optional | |
| targetObject | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| associationNames | String | - multiplicity is *<br>- unique<br>- optional | |
| entityTypeNames | String | - multiplicity is *<br>- unique<br>- optional | |
| distance | int | - multiplicity is 0..1<br>- unique<br>- optional | |
| selectedAttributesNames | String | - multiplicity is *<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.8. NameValueExpression

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

This datatype is extendable

### 3.1.2.8.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| attributeName | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| attributeValue | any | - multiplicity is 0..1<br>- unique<br>- optional | |
| operatorEnum | Operator | - multiplicity is 0..1<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.9. PolicyData

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.2.9.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| name | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| value | String | - multiplicity is 0..*<br>- unique<br>- optional | |

## 3.1.2.10. QueryFilter

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

org.tmforum.tip.inv.model.DataFilter

- Description:

An XPATH expression where the atomic terms represent either an attribute name or an attribute value

- Properties:

    This datatype is extendable

### 3.1.2.10.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| queryExpression | String | - multiplicity is 0..1<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.11. ScopeLevelSelector

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

    This datatype is extendable

### 3.1.2.11.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| nthLevel | int | - multiplicity is 0..1<br>- unique<br>- optional | Use to specify the level when levelEnum is set to: basetToNthLevel and NthLevelOnly |
| levelEnum | ScopeLevel | - multiplicity is 0..1<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.12. SimpleGraphQuery

- Type: Datatype Artifact
- Package: org.tmforum.tip.inv.model
- All super types:

    org.tmforum.tip.inv.model.BaseGraphQuery
- Properties:

    This datatype is abstract

    This datatype is extendable

### 3.1.2.12.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| startObject | objectName | - multiplicity is 1<br>- unique<br>- mandatory | |

| | | | |
|---|---|---|---|
| associationNames | String | - multiplicity is *<br>- unique<br>- optional | |
| selectedAttributesNames | String | - multiplicity is *<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 3.1.2.13. TemplateFilter

- Type: Datatype Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

org.tmforum.tip.inv.model.DataFilter

- Description:

A Template Object Instance is an Object Instance of a given Object Class where the attributes may not all be populated (even the mandatory ones).It is typically used to speed up Query operations by using matching capabilities at an Object Instance level.Thus, a given candidate Object Instance will match a Template Object Instance if, for all attributes that are populated in the Template Object Instance, each attribute has the same value as its corresponding attribute in the candidate Objet Instance. The meaning of “same value” is tied to the semantic of the equals() operation supported by the attribute.

- Properties:

This datatype is not extendable

### 3.1.2.13.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| template | EntityEnvelope | - multiplicity is 0..*<br>- unique<br>- passed by value<br>- optional | |

# 3.1.3. Notifications

## 3.1.3.1. EntityEnvelopeAVCNotification

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

org.tmforum.tip.common.notifications.AVCNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

- Description:

Notification about an Attribute Value Change.

- Properties:

This notification is mandatory

### 3.1.3.1.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| object | EntityEnvelope | - multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | |
| aliasNames | checkedCollection | - multiplicity is 0..1<br>- unique<br>- mandatory | The aliasNames attribute contains implimentation specific name value pairs for local alternative names for the Entity. |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

## 3.1.3.2. EntityEnvelopeOCNotification

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

org.tmforum.tip.common.notifications.OCNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

- Description:

Notificatin about an Object Create.

- Properties:

This notification is mandatory

### 3.1.3.2.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| object | EntityEnvelope | - multiplicity is 0..1<br>- unique<br>- passed by value<br>- optional | |
| aliasNames | checkedCollection | - multiplicity is 0..1<br>- unique<br>- mandatory | The aliasNames attribute contains implimentation specific name value pairs for local alternative names for the Entity. |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

## 3.1.3.3. EntityEnvelopeODelNotification

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

org.tmforum.tip.common.notifications.ODelNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

- Description:

Notification about an Object Delete

- Properties:

This notification is mandatory

### 3.1.3.3.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| object | EntityEnvelope | - multiplicity is 0..1<br>- unique<br>- optional | |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

## 3.1.3.4. EntityEnvelopeOCN

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- Description:

This is a notification generated from the entity org.tmforum.tip.inv.model.EntityEnvelope

- All super types:

org.tmforum.tip.common.notifications.OCNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

### 3.1.3.4.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| object | EntityEnvelope | - multiplicity is 0..1<br>- optional<br>- passed by value | Entity Envelope artifactAn Entity Envelope is an abstract model independent artifact used to encapsulate a model specific artifact.More precisely it is constituted of:• An Entity Identifier (Ref [1]) element, taken from JOSIF, which three components indicating The context to interpret the model specific artifact encapsulated in the content component the unique identifier of the specific artifact the type of the specific artifact• A model specific artifact that represents any kind of managed entity.For example in the case the context is the TMF Information Model (a.k.a. SID) it may represent a SID Entity, a SID Entity Specification. |
| aliasNames | CheckedCollection | - multiplicity is 0..1<br>- unique<br>- optional<br>- AVC enabled | The aliasNames attribute contains implementation specific name value pairs for local alternative names for the Entity. This is provided to pass more user friendly names for entities between systems or for debugging. The aliasNames attribute MUST NOT be used by an implimentation when comparing EntityIdentifiers. There is NO GUARANTEE that the contents of the aliasNames attribute is unique. |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

## 3.1.3.5. EntityEnvelopeODelN

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- Description:

This is a notification generated from the entity org.tmforum.tip.inv.model.EntityEnvelope

- All super types:

org.tmforum.tip.common.notifications.ODelNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

### 3.1.3.5.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| object | EntityEnvelope | - multiplicity is 0..1<br>- optional<br>- passed by value | Entity Envelope artifactAn Entity Envelope is an abstract model independent artifact used to encapsulate a model specific artifact.More precisely it is constituted of:• An Entity Identifier (Ref [1]) element, taken from JOSIF, which three components indicating The context to interpret the model specific artifact encapsulated in the content component the unique identifier of the specific artifact the type of the specific artifact• A model specific artifact that represents any kind of managed entity.For example in the case the context is the TMF Information Model (a.k.a. SID) it may represent a SID Entity, a SID Entity Specification. |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |

| | | | |
|---|---|---|---|
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

## 3.1.3.6. EntityEnvelopeResultAVCN

- Type: Event Artifact

- Package: org.tmforum.tip.inv.model

- Description:

This is a notification generated from the entity org.tmforum.tip.inv.model.EntityEnvelopeResult

- All super types:

org.tmforum.tip.common.notifications.AVCNotification

org.tmforum.tip.common.notifications.CommonNotification

org.tmforum.tip.internal.notifications.NotificationBase

### 3.1.3.6.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| reasonForFailure | String | - multiplicity is 0..1<br>- optional | |
| sourceTime | time | - multiplicity is 0..1<br>- unique<br>- mandatory | The time at which the event was reported by the source system (NE, EMS or OS). |
| objectId | EntityIdentifier | - multiplicity is 0..1<br>- unique<br>- mandatory | The identifier of the object associated with the event, as internal opaque identifier. |
| objectType | String | - multiplicity is 0..1<br>- unique<br>- mandatory | The type (class) of the object associated with the event. This attribute is needed to allow simple notification filtering based on the object type. |
| extensionInfo | Any | - multiplicity is 0..1<br>- unique<br>- optional | A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces. |

# 3.1.4. Enumerations

## 3.1.4.1. AttributeModificationPatternOperator

- Type: Enumeration Artifact

- Package: org.tmforum.tip.inv.model

- Properties:

### 3.1.4.1.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| SET | String | value is "SET" | |
| REMOVEALLVALUES | String | value is "REMOVEALLVALUES" | |
| REMOVEVALUES | String | value is "REMOVEVALUES" | |
| SETTODEFAULT | String | value is "SETTODEFAULT" | |
| ADDVALUES | String | value is "ADDVALUES" | |

## 3.1.4.2. AttributeValuePolicy

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.4.2.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| Strict | String | value is "Strict " | |
| Loose | String | value is "Loose" | |
| NotDefined | String | value is "NotDefined" | |

## 3.1.4.3. ExecutionMode

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

This enumeration is not extendable

### 3.1.4.3.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| BESTEFFORT | String | value is "BESTEFFORT" | |
| ATOMIC | String | value is "ATOMIC" | |

## 3.1.4.4. MandatoryAssociationPolicy

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.4.4.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| Strict | String | value is "Strict" | |
| Loose | String | value is "Loose " | |
| NotDefined | String | value is "NotDefined" | |

# 3.1.4.5. ModelConformanceValidationPolicy

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.4.5.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| ValidationPerRequest | String | value is "ValidationPerRequest " | |
| ValidationOnDemand | String | value is "ValidationOnDemand" | |

# 3.1.4.6. NamingPolicy

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.4.6.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| ClientControl | String | value is "ClientControl " | |
| ServerControl | String | value is "ServerControl" | |
| Both | String | value is "Both" | |

# 3.1.4.7. OperationTargetScope

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

### 3.1.4.7.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| RECURSIVE | String | value is "RECURSIVE" | |
| BASEOBJECTSONLY | String | value is "BASEOBJECTSONLY" | |

# 3.1.4.8. Operator

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

    This enumeration is extendable

#### 3.1.4.8.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| equal | String | value is "equal" | |
| less | String | value is "less" | |
| lessOrEqual | String | value is "lessOrEqual" | |
| greater | String | value is "greater" | |
| greaterOrEqual | String | value is "greaterOrEqual" | |
| like | String | value is "like" | |
| contains | String | value is "contains" | Substring matching for String based attributes |
| isElementOf | String | value is "isElementOf" | Test presence of element in an Array |
| isPresent | String | value is "isPresent" | |

### 3.1.4.9. ResultIndicator

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Properties:

#### 3.1.4.9.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| ALL | String | value is "ALL" | |
| BASEOBJECTSONLY | String | value is "BASEOBJECTSONLY" | |

### 3.1.4.10. ScopeLevel

- Type: Enumeration Artifact
- Package: org.tmforum.tip.inv.model
- Description:

BASE_NTH_LEVEL:specifies that all subordinate Managed Objects that are at "BASE_NTH_LEVEL" distance from the root Managed Object are of interest;where 0 is the root Managed Object. BASE_TO_NTH_LEVEL:specifies that the root Managed Object and all its subordinates down to and including level "BASE_TO_NTH_LEVEL" are of interest;where 0 is the root Managed Object. BASE_ALL:specifies that the root Managed Object and all its subordinates are of interest.

- Properties:

    This enumeration is extendable

#### 3.1.4.10.1. Literals

| name | datatype | properties | description |
|---|---|---|---|
| baseToNthLevel | String | value is "baseToNthLevel" | |
| allLevels | String | value is "allLevels" | |
| NthLevelOnly | String | value is "NthLevelOnly" | |

# 3.1.5. Exceptions

## 3.1.5.1. CreateException

- Type: Exception Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

  org.tmforum.tip.common.exceptions.CommonException

  org.tmforum.tip.internal.exceptions.ExceptionBase

- Properties:

#### 3.1.5.1.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| reason | String | - multiplicity is 0..1<br>- unique<br>- optional | Free string to define the reason the exception was raised. It should be a short text. Might reference the step in the use case where this exception occurs, if a detailed use case was made for this operation. |
| details | String | - multiplicity = 0..1<br>- unique<br>- optional | The more specific details about the exception. Can also be a stack trace. |

## 3.1.5.2. DeleteException

- Type: Exception Artifact

- Package: org.tmforum.tip.inv.model

- All super types:

  org.tmforum.tip.common.exceptions.CommonException

  org.tmforum.tip.internal.exceptions.ExceptionBase

- Properties:

#### 3.1.5.2.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| reason | String | - multiplicity is 0..1<br>- unique<br>- optional | Free string to define the reason the exception was raised. It should be a short text. Might reference the step in the use case where this exception occurs, if a detailed use case was made for this operation. |
| details | String | - multiplicity = 0..1<br>- unique<br>- optional | The more specific details about the exception. Can also be a stack trace. |

## 3.1.5.3. ModifyException

- Type: Exception Artifact
- Package: org.tmforum.tip.inv.model
- All super types:
  - org.tmforum.tip.common.exceptions.CommonException
  - org.tmforum.tip.internal.exceptions.ExceptionBase
- Properties:

### 3.1.5.3.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| reason | String | - multiplicity is 0..1<br>- unique<br>- optional | Free string to define the reason the exception was raised. It should be a short text. Might reference the step in the use case where this exception occurs, if a detailed use case was made for this operation. |
| details | String | - multiplicity = 0..1<br>- unique<br>- optional | The more specific details about the exception. Can also be a stack trace. |

## 3.1.5.4. ValidationException

- Type: Exception Artifact
- Package: org.tmforum.tip.inv.model
- All super types:
  - org.tmforum.tip.common.exceptions.AlreadyInPostCondition
  - org.tmforum.tip.common.exceptions.CommonException
  - org.tmforum.tip.internal.exceptions.ExceptionBase
- Properties:

### 3.1.5.4.1. Attributes

| name | datatype | properties | description |
|---|---|---|---|
| reason | String | - multiplicity is 0..1<br>- unique<br>- optional | Free string to define the reason the exception was raised. It should be a short text. Might reference the step in the use case where this exception occurs, if a detailed use case was made for this operation. |
| details | String | - multiplicity = 0..1<br>- unique<br>- optional | The more specific details about the exception. Can also be a stack trace. |

# 4. Service Interfaces

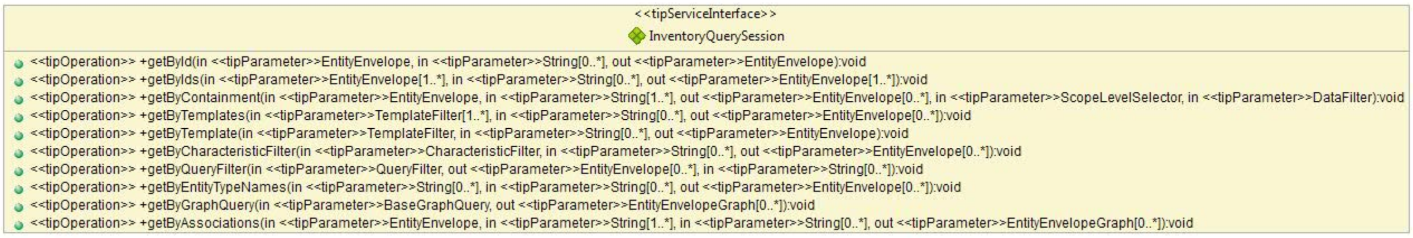
Service interfaces available from TIP Generic Inventory Management Interface model:
- InventoryQuerySession
- InventoryUpdateSession

## 4.1. InventoryQuerySession

- Type: Session Artifact (Service Interface)
- Package: org.tmforum.tip.inv.model
- Description:

Service Interface containing all the Query operations.This service interface supports the various notfications, both the specific Invnetory notifications and the Heartbeat notification.

- Operations exposed:
    - getById
    - getByIds
    - getByContainment
    - getByTemplates
    - getByTemplate
    - getByCharacteristicFilter
    - getByQueryFilter
    - getByEntityTypeNames
    - getByGraphQuery
    - getByAssociations
- Common Operations
- Emitted events:
    - org.tmforum.tip.inv.model.EntityEnvelopeAVCNotification
    - org.tmforum.tip.inv.model.EntityEnvelopeOCNotification
    - org.tmforum.tip.inv.model.EntityEnvelopeODelNotification
- Properties:
    - This service interface is mandatory



```
<<tipServiceInterface>>
InventoryQuerySession
<<tipOperation>> +getById(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByIds(in <<tipParameter>>EntityEnvelope[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[1..*]):void
<<tipOperation>> +getByContainment(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>ScopeLevelSelector, in <<tipParameter>>DataFilter):void
<<tipOperation>> +getByTemplates(in <<tipParameter>>TemplateFilter[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByTemplate(in <<tipParameter>>TemplateFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope):void
<<tipOperation>> +getByCharacteristicFilter(in <<tipParameter>>CharacteristicFilter, in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByQueryFilter(in <<tipParameter>>QueryFilter, out <<tipParameter>>EntityEnvelope[0..*], in <<tipParameter>>String[0..*]):void
<<tipOperation>> +getByEntityTypeNames(in <<tipParameter>>String[0..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelope[0..*]):void
<<tipOperation>> +getByGraphQuery(in <<tipParameter>>BaseGraphQuery, out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
<<tipOperation>> +getByAssociations(in <<tipParameter>>EntityEnvelope, in <<tipParameter>>String[1..*], in <<tipParameter>>String[0..*], out <<tipParameter>>EntityEnvelopeGraph[0..*]):void
```

### 4.1.1. getById

- Type: Operation
- Description:

Query a single Managed Object knowing its identifier.

- Properties:

 This operation is extendable

 This operation is mandatory

- Return:

 void

- Exceptions:

 EntityNotFound

 AccessDenied

 CommunicationLoss

 InternalError

 InvalidInput

 NotImplemented

 UnableToComply

#### 4.1.1.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| targetObject | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by id<br>- mandatory | Identifier of the Managed Object of interest |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- mandatory | Llist of Attribute Names. |
| object | EntityEnvelope | - output parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | Entity Envelope containing the identified ManagedObject with only the valued attributes corresponding to the list of Attribute. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.2. getByIds

- Type: Operation
- Description:

Query several Managed Objects knowing their Identifiers

- Properties:

 This operation is extendable

 This operation is optional

- Return:

 void

- Exceptions:

 CapacityExceeded

 AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.1.2.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| targetedObjects | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by id<br>- mandatory | The identifiers of the Managed Objects of interest |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- mandatory | Llist of Attribute Names. |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by value<br>- mandatory | Entity Envelopes containing the identified ManagedObject with only the valued attributes corresponding to the list of Attribute. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.3. getByContainment

- Type: Operation

- Description:

Query several Managed Objects in the same containment tree

- Properties:

 This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

EntityNotFound

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.1.3.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| baseObject | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by id<br>- mandatory | The identifier of a single Managed Object (the root of the subtree) |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | Llist of Attribute Names. |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Entity Envelopes Collection containing the Managed Objects of interest with only the valued attributes corresponding to the list of Attribute Names. |
| level | ScopeLevelSelector | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | A parameter indicating how many levels, in the containment tree, to select from from the root Managed Object |
| filter | DataFilter | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- optional | this filter may be a Query Filter, a Template Filter or a Characteristics Filter. the Target System uses it to select from the Managed Objects already scoped in the containment tree only the ones matching the filter (called the Managed Objects of interest). |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.4. getByTemplates

- Type: Operation

- Description:

Query several Managed Objects matching any Template from a List

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

This operation can emit events

- Return:

void

- Exceptions:

CapacityExceeded

FilterNotSupported

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.1.4.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| templates | TemplateFilter | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | One or several independent lists of valued attributes, each list belonging to a valid Object Class of the Information Model supported by the Target System (called the templates); |

| | | | |
|---|---|---|---|
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | Llist of Attribute Names. |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Entity Envelopes Collection containing the Managed Objects of interest with only the valued attributes corresponding to the list of Attribute Names. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.5. getByTemplate

- Type: Operation

- Description:

Query several Managed Objects matching a Template

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is mandatory

- Return:

void

- Exceptions:

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.1.5.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| template | TemplateFilter | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | |
| selectedAttributeNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | Llist of Attribute Names. |
| returnedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..1<br>- unique<br>- passed by value<br>- bulk potential<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.6. getByCharacteristicFilter

- Type: Operation

- Description:

Query several Managed Objects matching a Characteristics Filter

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

This operation can emit events

- Return:

void

- Exceptions:

CapacityExceeded

FilterNotSupported

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.1.6.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| filter | CharacteristicFilter | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | Logical expression where atomic terms represent valued attributes; |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | Llist of Attribute Names. |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Entity Envelopes Collection containing the Managed Objects of interest with only the valued attributes corresponding to the list of Attribute Names. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.7. getByQueryFilter

- Type: Operation

- Description:

Query several Managed Objects matching a Query Filter

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

FilterNotSupported

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.1.7.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| filter | QueryFilter | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | An XPATH expression where the atomic terms represent either an attribute name or an attribute value |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.8. getByEntityTypeNames

- Type: Operation

- Description:

Query by Entity Type Names

- Properties:

This operation is extendable

This operation is mandatory

- Return:

void

- Exceptions:

CapacityExceeded

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.1.8.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| entityTypes | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- mandatory | List of “Entity Type Names |
| selectedAttributesNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | Llist of Attribute Names. |
| objects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Entity Envelopes Collection containing the Managed Objects of interest with only the valued attributes corresponding to the list of Attribute Names. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.1.9. getByGraphQuery

- Type: Operation

- Description:

Query a Graph of Managed Objects based on a Graph Query.

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

EntityNotFound

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.1.9.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| graphQuery | BaseGraphQuery | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | |
| graph | EntityEnvelopeGraph | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Entity Envelopes Collection containing the Managed Objects of interest. |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

### 4.1.10. getByAssociations

- Type: Operation
- Description:

Query a Graph of Managed Objects based on Association Selection.

- Properties:

This operation uses iterator bulk transfer pattern

This operation is extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

EntityNotFound

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.1.10.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| origin | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by id<br>- mandatory | |
| associationNames | String | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | |
| attributeNames | String | - input parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | Llist of Attribute Names. |
| graph | EntityEnvelopeGraph | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | |
| extensionInfo | Any | - multiplicity is 0..1<br>- input/output parameter<br>- optional | a generic list of any type of elements. Used for vendor extensions or loose element encapsulation from other namespaces. |

## 4.2. InventoryUpdateSession

- Type: Session Artifact (Service Interface)
- Package: org.tmforum.tip.inv.model
- Operations exposed:

getPolicies

validateByIds

validateByTemplateFilters

createByValue

createByValues

createByTemplate

createMultipleByTemplate

deleteByIds

deleteByTemplateFilters

modifyByTemplate

modifyByValue

modifyById

modifyByIds

modifyByAttributePatterns

modifyByTemplateFilters

modifyByValues

- Common Operations
- Properties:

This service interface is optional

## 4.2.1. getPolicies

- Type: Operation
- Description:

The Interface supports the capability for a Requesting System to query the Target System about which all policies related to the Update feature

- Properties:

This operation is not extendable

This operation is mandatory

- Return:

void

- Exceptions:

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.1.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| policies | PolicyData | - output parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | |

# 4.2.2. validateByIds

- Type: Operation

- Description:

Validate Multiple Objects knowing their Identifiers

- Properties:

 This operation is not extendable

 This operation is optional

- Return:

 void

- Exceptions:

 ValidationException

 AccessDenied

 CommunicationLoss

 InternalError

 InvalidInput

 NotImplemented

 UnableToComply

### 4.2.2.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| baseObjectsToValidate | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by id<br>- mandatory | the Identifiers of the Objects (Base Objects) to check for validation. |
| scope | OperationTargetScope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | Operation Scope |
| objectsChecked | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Objects that were validated |
| objectsNotChecked | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Objects that do not exist |

# 4.2.3. validateByTemplateFilters

- Type: Operation

- Description:

Validate Multiple Objects exactly matching anyTemplateFilter from a List

- Properties:

 This operation uses iterator bulk transfer pattern

 This operation is not extendable

 This operation is optional

- Return:

  void

- Exceptions:

  ValidationException

  AccessDenied

  CommunicationLoss

  InternalError

  InvalidInput

  NotImplemented

  UnableToComply

#### 4.2.3.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| templates | TemplateFilter | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | |
| scope | OperationTargetScope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | |
| objectsChecked | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | |

## 4.2.4. createByValue

- Type: Operation

- Description:

Create a Single Object

- Properties:

  This operation is not extendable

  This operation is mandatory

- Return:

  void

- Exceptions:

  AtomicTransactionFailure

  Duplicate

  CreateException

  AccessDenied

  CommunicationLoss

  InternalError

  InvalidInput

  NotImplemented

  UnableToComply

### 4.2.4.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| objectToCreate | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | ObjectToCreate Entity Envelope |
| templateObjectReference | EntityEnvelope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- passed by id<br>- optional | an Entity Identifier referring to an Object that must exist in the Target System |
| isAutoNamingRequested | boolean | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'false'<br>- optional | Naming Boolean parameter |
| createdObject | EntityEnvelope | - output parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | A copy of the created Object encapsulated in an Entity Envelope. |

## 4.2.5. createByValues

- Type: Operation

- Description:

Create Multiple Objects of Multiple Classes

- Properties:

This operation is not extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

CreateException

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.5.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| objectsToCreate | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by value<br>- mandatory | Objects To Create A list of Entity Envelopes |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | this parameter indicates whether the execution should be BestEffort or Atomic |

| | | | |
|---|---|---|---|
| isAutoNamingRequested | boolean | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'false'<br>- optional | Naming boolean parameter |
| createdObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- optional | list of Entity Envelopes, where each item contains a copy of a created Object encapsulated in an Entity Envelope |
| failedObjects | FailedObjectResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- optional | list of Failed Objects by Id |

## 4.2.6. createByTemplate

- Type: Operation

- Description:

Create Multiple Objects of the Same Classby Value

- Properties:

This operation is not extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

CreateException

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.6.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| objectsToCreate | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by value<br>- mandatory | Entities to be Created |
| templateObjectValue | EntityEnvelope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- passed by value<br>- optional | Template for Created Entities |
| isAutoNamingRequested | boolean | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'false'<br>- optional | Auto Naming Parameter |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | Execution Mode |

| | | | |
|---|---|---|---|
| createdObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Created Objects |
| failedObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Failed Objects |

# 4.2.7. createMultipleByTemplate

- Type: Operation

- Description:

Create n Objects of the Same Classby Value

- Properties:

This operation is not extendable

This operation is optional

- Return:

void

- Exceptions:

CapacityExceeded

CreateException

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.7.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| numberOfObjectsToCreate | int | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | An integer value indicating how many Objects must be created. |
| objectClassName | String | - input parameter<br>- multiplicity is 1<br>- unique<br>- mandatory | The name of the Object Class of the Objects to create. |
| referenceTemplateValue | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | an Entity Envelope used as a template to create the entities |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | this parameter indicates whether the execution should be BestEffort or Atomic |
| superiorObjectId | EntityEnvelope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- passed by id<br>- optional | the corresponding Object will be the superior of all the Objects to create (empty in the case the Objects to create are at the root level |
| createdObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Copies of Created Objects |

|  |  |  |  |
|---|---|---|---|
| failedObjects | FailedObjectResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- mandatory | Failed Objects |

# 4.2.8. deleteByIds

- Type: Operation

- Description:

Delete Multiple Objects knowing their Identifiers

- Properties:

This operation is not extendable

This operation is mandatory

- Return:

void

- Exceptions:

DeleteException

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.8.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| baseObjectsToDelete | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by id<br>- mandatory | |
| scope | OperationTargetScope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | |
| resultIndicator | ResultIndicator | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | |
| deletedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by id<br>- mandatory | |
| failedObjects | FailedObjectResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- mandatory | |

# 4.2.9. deleteByTemplateFilters

- Type: Operation

- Description:

Delete Multiple Objects exactly matching any TemplateFilter from a List

- Properties:

 This operation uses iterator bulk transfer pattern

This operation is not extendable

This operation is optional

- Return:

void

- Exceptions:

DeleteException

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.9.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| templateFilter | TemplateFilter | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | Templates used for Targeting entities |
| scope | OperationTargetScope | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | Operation Scope |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | Execution Mode |
| resultIndicator | ResultIndicator | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BASEOBJECTSONLY'<br>- optional | Result Indicator |
| failedAndDeletdObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- optional | Failed and Deleted Objects in Iterator |

# 4.2.10. modifyByTemplate

- Type: Operation

- Description:

Modify Multiple Objects of the Same Classby Value knowing their Identifiers

- Properties:

This operation is not extendable

This operation is optional

- Return:

void

- Exceptions:

ModifyException

CapacityExceeded

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.10.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| modificationObjects | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by value<br>- mandatory | A list of Entity Envelopes; |
| templateObjectValue | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | Entity Envelope with ValueObject |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | Execution Mode |
| modifiedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | the modified Objects encapsulated in an Entity Envelopes |
| failedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Failed Objects |

## 4.2.11. modifyByValue

- Type: Operation

- Description:

Modify a Single Object knowing its Identifier

- Properties:

This operation is not extendable

This operation is mandatory

- Return:

void

- Exceptions:

ModifyException

EntityNotFound

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.2.11.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| modificationObject | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | Object to be Modified |
| modifiedObject | EntityEnvelope | - output parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | Modified Object |

### 4.2.12. modifyById

- Type: Operation

- Description:

Modify a Single Object with Modification Patternsknowing its Identifier

- Properties:

This operation is not extendable

This operation is mandatory

- Return:

void

- Exceptions:

ModifyException

EntityNotFound

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

#### 4.2.12.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| identifierOfObjectToModify | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by id<br>- mandatory | The Identifier of the Object to modify; |
| attributeModificationPatterns | AttributeModificationPattern | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | A set of Attribute Modification Patterns |

| | | | |
|---|---|---|---|
| modifiedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | The modified objects by value |

## 4.2.13. modifyByIds

- Type: Operation

- Description:

Modify Multiple Objects with Modification Patternsknowing their Identifiers

- Properties:

This operation is not extendable

This operation is mandatory

- Return:

void

- Exceptions:

ModifyException

CapacityExceeded

AccessDenied

CommunicationLoss

InternalError

InvalidInput

NotImplemented

UnableToComply

### 4.2.13.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| identifiersOfObjectsToModify | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by id<br>- mandatory | A list of Identifiers (mandatory):this parameter represents the Identifiers of the Objects to modify |
| attributeModificationPatterns | AttributeModificationPattern | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | A set of Attribute Modification Patterns |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | Execution Mode |
| modifiedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | the modified Objects encapsulated in Entity Envelopes array |
| failedObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | failedObjects |

## 4.2.14. modifyByAttributePatterns

- Type: Operation

- Description:

Modify Multiple Objects with Modification Patternsexactly matching any TemplateFilter from a List

- Properties:

 This operation is not extendable

 This operation is optional

- Return:

 void

- Exceptions:

 ModifyException

 CapacityExceeded

 AccessDenied

 CommunicationLoss

 InternalError

 InvalidInput

 NotImplemented

 UnableToComply

#### 4.2.14.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| templateFilters | TemplateFilter | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | A list of Template Filters |
| attributeModificationPatterns | AttributeModificationPattern | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | A set of Attribute Modification Patterns |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 1<br>- unique<br>- default value is 'BESTEFFORT'<br>- mandatory | Execution Mode |
| failedAndModifiedObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Failed and Modified Objects using bulk transfer |

## 4.2.15. modifyByTemplateFilters

- Type: Operation

- Description:

Modify Multiple Objects of the Same Classby Value and exactly matching any TemplateFilter from a List

- Properties:

 This operation uses iterator bulk transfer pattern

 This operation is not extendable

 This operation is optional

- Return:

 void

- Exceptions:

    ModifyException

    CapacityExceeded

    AccessDenied

    CommunicationLoss

    InternalError

    InvalidInput

    NotImplemented

    UnableToComply

#### 4.2.15.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| templateFilters | TemplateFilter | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- mandatory | A list of Templates. |
| referenceTemplate | EntityEnvelope | - input parameter<br>- multiplicity is 1<br>- unique<br>- passed by value<br>- mandatory | A ValueObject |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | Execution Mode |
| failedAndModifiedObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- bulk potential<br>- mandatory | Failed and Modified Objects using bulk transfer |

## 4.2.16. modifyByValues

- Type: Operation
- Description:

Modify Multiple Objects of Multiple Classes

- Properties:

    This operation is not extendable

    This operation is optional

- Return:

    void

- Exceptions:

    ModifyException

    CapacityExceeded

    AccessDenied

    CommunicationLoss

    InternalError

    InvalidInput

    NotImplemented

    UnableToComply

#### 4.2.16.1. Arguments

| name | datatype | properties | description |
|---|---|---|---|
| modifcationObjects | EntityEnvelope | - input parameter<br>- multiplicity is 1..*<br>- unique<br>- passed by value<br>- mandatory | |
| executionMode | ExecutionMode | - input parameter<br>- multiplicity is 0..1<br>- unique<br>- default value is 'BESTEFFORT'<br>- optional | |
| modifiedObjects | EntityEnvelope | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Copies of Modified Objects |
| failedObjects | EntityEnvelopeResult | - output parameter<br>- multiplicity is 0..*<br>- unique<br>- passed by value<br>- mandatory | Objects that failed |