Integrate Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0

Committee Specification 01

31 January 2013

Specification URIs
This version:

http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/com-ics-v1.0.cs01.doc (Authoritative)
http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/com-ics-v1.0.cs01.html
http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/com-ics-v1.0.cs01.pdf

Previous version:

http://www.oasis-open.org/committees/download.php/46823/com-ics-v1.0-csprod04.zip

Latest version:

http://docs.oasis-open.org/icom/icom-ics/v1.0/com-ics-v1.0.doc (Authoritative)
http://docs.oasis-open.org/icom/icom-ics/v1.0/com-ics-v1.0.html
http://docs.oasis-open.org/icom/icom-ics/v1.0/com-ics-v1.0.pdf

Technical Committee:
OASIS Integrated Collaboration Object Model for Interoperable Collaboration Services (ICOM) TC

Chairs:
Eric S. Chan (eric.s.chan@oracle.com), Oracle
Kenneth P. Baclawski (kenb@ccs.neu.edu), Northeastern University

Editors:
Eric S. Chan (eric.s.chan@oracle.com), Oracle
Patrick Durusau (patrick@durusau.net), Individual

Additional artifacts:
This prose specification is one component of a Work Product which also includes:
- XML schemas: http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/schemas/

Declared XML namespaces:
http://docs.oasis-open.org/ns/icom/core/201008
http://docs.oasis-open.org/ns/icom/accesscontrol/201008
http://docs.oasis-open.org/ns/icom/metadata/201008
http://docs.oasis-open.org/ns/icom/content/201008
http://docs.oasis-open.org/ns/icom/document/201008
http://docs.oasis-open.org/ns/icom/message/201008
http://docs.oasis-open.org/ns/icom/presence/201008
http://docs.oasis-open.org/ns/icom/contact/201008
http://docs.oasis-open.org/ns/icom/calendar/201008
http://docs.oasis-open.org/ns/icom/task/201008
http://docs.oasis-open.org/ns/icom/forum/201008
http://docs.oasis-open.org/ns/icom/conference/201008
Abstract:
The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services defines a framework for integrating a broad range of domain models for collaboration activities in an integrated and interoperable collaboration environment.

The framework is not intended to prescribe how applications or services conforming to its model implement, store, or transport the data for objects. It is intended as a basis for integrating a broad range of collaboration objects to enable seamless transitions across collaboration activities. This enables applications to maintain a complete thread of conversations across multiple collaboration activities.

The model integrates a broad range of collaboration activities, by encompassing and improving on a range of models which are part of existing standards and technologies. The model is modular to allow extensibility. The core concepts, metadata concepts, and their relations are included in the Core, while the specific concepts and relations for each area of collaboration activities are defined in separate extension modules.

Status:
This document was last revised or approved by the OASIS Integrated Collaboration Object Model for Interoperable Collaboration Services (ICOM) TC on the above date. The level of approval is also listed above.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using “Send A Comment” button on the Technical Committee's web page at http://www.oasis-open.org/committees/icom/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/icom/ipr.php).

Citation format:
When referencing this specification the following citation format should be used:

[ICOM-ics-v1.0]
Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. 31 January 2013. OASIS Committee Specification 01.
http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/icom-ics-v1.0-cs01.html.
Notices

Copyright © OASIS Open 2013. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.
Table of Contents

1 Introduction ............................................................................................................................................. 11
  1.1 Terminology ...................................................................................................................................... 12
  1.2 Normative References ......................................................................................................................... 12
  1.3 Non-Normative References ................................................................................................................ 12
2 Modeling Language ........................................................................................................................... 14
  2.1 Introduction ...................................................................................................................................... 14
  2.2 Class Definition Grammar ................................................................................................................. 14
  2.3 Property Definition Grammar .......................................................................................................... 16
  2.4 Namespaces ...................................................................................................................................... 19
3 Core Model .......................................................................................................................................... 20
  3.1 Main Branch ................................................................................................................................... 20
    3.1.1 Entity and Top-Level Subclasses ............................................................................................... 20
    3.1.2 Identifiable .................................................................................................................................. 20
    3.1.3 Parental ....................................................................................................................................... 21
    3.1.4 Extent .......................................................................................................................................... 22
    3.1.5 Entity .......................................................................................................................................... 23
    3.1.6 Overview of Scope, Subject, and Artifact Branches .................................................................... 27
  3.2 Scope Branch ................................................................................................................................... 28
    3.2.1 Scope and Top-Level Subclasses ............................................................................................... 28
    3.2.2 Scope .......................................................................................................................................... 29
    3.2.3 Community ................................................................................................................................. 31
    3.2.4 Space .......................................................................................................................................... 33
  3.3 Subject Branch .................................................................................................................................. 35
    3.3.1 Subject and Top-Level Subclasses ............................................................................................. 35
    3.3.2 Subject ....................................................................................................................................... 35
    3.3.3 Group ......................................................................................................................................... 37
    3.3.4 Actor .......................................................................................................................................... 39
    3.3.5 Person ......................................................................................................................................... 41
    3.3.6 Resource ..................................................................................................................................... 45
    3.3.7 ResourceType ............................................................................................................................... 47
    3.3.8 ResourceTypeEnum ..................................................................................................................... 48
    3.3.9 ResourceBookingRule ............................................................................................................... 49
    3.3.10 ResourceBookingRuleEnum .................................................................................................... 50
  3.4 Artifact Branch .................................................................................................................................. 51
    3.4.1 Artifact and Top-Level Subclasses ............................................................................................. 51
    3.4.2 Item ............................................................................................................................................. 51
    3.4.3 SpaceItem ................................................................................................................................. 53
    3.4.4 Container .................................................................................................................................... 53
    3.4.5 FolderContainer .......................................................................................................................... 54
    3.4.6 Artifact ....................................................................................................................................... 55
    3.4.7 Folder ......................................................................................................................................... 58
    3.4.8 HeterogeneousFolder ................................................................................................................ 59
  3.5 Access Control Model ....................................................................................................................... 61
3.5.1 Accessor ............................................................................................................. 61
3.5.2 Owner .................................................................................................................. 61
3.5.3 RoleDefinition ...................................................................................................... 62
3.5.4 Role ...................................................................................................................... 63
3.5.5 Privilege ............................................................................................................... 65
3.5.6 PrivilegeEnum ..................................................................................................... 66
3.5.7 AccessControlList ............................................................................................. 67
3.5.8 AccessControlEntry .......................................................................................... 68
3.5.9 AccessType .......................................................................................................... 69
3.5.10 AccessTypeEnum ............................................................................................... 70
3.6 Metadata Model ...................................................................................................... 71
3.6.1 ClassDefinition .................................................................................................... 71
3.6.2 Stereotype ........................................................................................................... 74
3.6.3 StereotypeEnum .................................................................................................. 75
3.6.4 PropertyDefinition .............................................................................................. 76
3.6.5 Property ............................................................................................................... 79
3.6.6 PropertyChoiceType ........................................................................................... 81
3.6.7 PropertyType ...................................................................................................... 82
3.6.8 PropertyTypeEnum .............................................................................................. 83
3.6.9 Updatability ......................................................................................................... 84
3.6.10 UpdatabilityEnum ............................................................................................ 85
3.6.11 Cardinality ......................................................................................................... 86
3.6.12 CardinalityEnum ............................................................................................... 86
3.6.13 Marker and Subclasses ...................................................................................... 87
3.6.14 Marker ............................................................................................................... 88
3.6.15 Category ............................................................................................................ 89
3.6.16 CategoryApplication ......................................................................................... 91
3.6.17 Tag ..................................................................................................................... 92
3.6.18 TagApplication ................................................................................................. 94
3.6.19 RelationshipBondable ...................................................................................... 95
3.6.20 RelationshipDefinition ..................................................................................... 96
3.6.21 Relationship ...................................................................................................... 98
3.7 Common Concepts ................................................................................................. 100
3.7.1 Addressable ....................................................................................................... 100
3.7.2 EntityAddress .................................................................................................... 101
3.7.3 Participant .......................................................................................................... 102
3.7.4 Priority ............................................................................................................... 104
3.7.5 PriorityEnum ..................................................................................................... 105
3.7.6 DateTimeResolution ............................................................................................ 105
3.7.7 DateTimeResolutionEnum ................................................................................... 106
3.7.8 TimeZone ............................................................................................................. 107
3.7.9 Location ............................................................................................................... 108
3.7.10 GeoCoordinates ............................................................................................... 110
4 Extension Modules .................................................................................................. 112
4.1 Overview of Extension Modules ........................................................................... 112
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Entity and Top-Level Abstract Classes</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Entity Class Diagram</td>
<td>27</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Scope, Subject, and Artifact Branches</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Scope Branch</td>
<td>28</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Scope Class Diagram</td>
<td>31</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Community Class Diagram</td>
<td>33</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Space Class Diagram</td>
<td>34</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Subject Branch</td>
<td>35</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Subject Class Diagram</td>
<td>37</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Group and Actor Class Diagram</td>
<td>39</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Person Class Diagram</td>
<td>45</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Resource Class Diagram</td>
<td>47</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Artifact Branch</td>
<td>51</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Artifact Class Diagram</td>
<td>58</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Heterogeneous Folder Class Diagram</td>
<td>60</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Role Definition and Role Class Diagram</td>
<td>65</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Access Control List Class Diagram</td>
<td>71</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Class Definition and Property Definition Class Diagram</td>
<td>74</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Property Definition and Property Class Diagram</td>
<td>81</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Marker Branch</td>
<td>87</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Marker Class Diagram</td>
<td>89</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Category and Category Application Class Diagram</td>
<td>91</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Tag and Tag Application Class Diagram</td>
<td>94</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Relationship Class Diagram</td>
<td>100</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Containers of Collaboration Activities</td>
<td>112</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Composite Content Class Diagram</td>
<td>116</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Document, Version Series, and Version Class Diagram</td>
<td>134</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Wiki Page Class Diagram</td>
<td>136</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Unified Message Class Diagram</td>
<td>150</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Instant Message Class Diagram</td>
<td>153</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Instant Message Feed and Connection Class Diagram</td>
<td>158</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Presence Class Diagram</td>
<td>163</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Presence Contact Method and Instant Message Connection Class Diagram</td>
<td>169</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Address Book Class Diagram</td>
<td>174</td>
</tr>
<tr>
<td>Figure 35</td>
<td>Person Contact Class Diagram</td>
<td>179</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Calendar Class Diagram</td>
<td>181</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Occurrence Series Class Diagram</td>
<td>186</td>
</tr>
<tr>
<td>Figure 38</td>
<td>Occurrence Class Diagram</td>
<td>191</td>
</tr>
<tr>
<td>Figure 39</td>
<td>Free Busy Class Diagram</td>
<td>204</td>
</tr>
<tr>
<td>Figure 40</td>
<td>Task List Class Diagram</td>
<td>207</td>
</tr>
</tbody>
</table>
Figure 41: Task Class Diagram. ................................................................. 211
Figure 42: Forum Class Diagram. ............................................................ 221
Figure 43: Conference Class Diagram. ..................................................... 230
1 Introduction

The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services specification defines a framework for integrating a broad range of domain model for collaboration activities in an interoperable collaboration environment. The standard promotes an integrated user experience with seamless transitions across collaboration activities. It enables applications to support continuity of conversations across diverse collaboration activities. For example, applications can aggregate conversation threads in email with other conversations on the same topic in instant message, over the phone or via real-time conferencing, by discussion threads in community forum, weblog or micro blog, and activity stream of participants from all channels.

The specification defines a core model and a set of extension modules. The core model (Section 3) defines the classes (Section 3.1 Main Branch) that bring together the model of directory (Section 3.2 Scope Branch), identity management (Section 3.3 Subject Branch), and content management (Section 3.4 Artifact Branch) in a framework with a common access control model (Section 3.5) and metadata model (Section 3.6). The extension modules in Section 4 extend the artifact and folder model of Artifact Branch (Section 3.4) to define the specialized model for different collaboration activities. The range of collaboration model includes content sharing and co-creation, asynchronous communication, instant communication, presence awareness, moderated group discussion, time management, coordination, real-time interaction, etc.

The Subject and Artifact branches support separation of concerns for user administration and content management. Subject branch includes the model of actors, groups of actors, and role assignment of actors. Actors, groups, and roles typically appear as the subject in the (subject, privilege, object) triples of an access control model. The Artifact branch includes the model of content and metadata produced by actors. The Scope branch includes the model of communities and spaces that contain subjects and artifacts. Communities and spaces join the subjects and artifacts in a role-based access control model where a role is assigned to an actor in a specific scope. Thus Scope, Subject, and Artifact form a framework for applications to integrate and interoperate with directory, identity management, content management, and collaboration services.

The model specified in ICOM is part of existing standards and technologies, several of which are referenced in Section 1.3 Non-Normative References. The model is modular and extensible, with common concepts, metadata concepts, and their relations provided in the Core, while the specific concepts and relations for each area of collaboration activities defined in separate extension modules.

ICOM core model encompasses LDAP Directory Information Models [RFC4512]. The extension modules integrate models from Content Management Interoperability Services [CMIS], Java Content Repository API [JCR 2.0], Web Distributed Authoring and Versioning (WebDAV) [RFC4918], Internet Message Access Protocol (IMAP) [RFC2119], Simple Mail Transfer Protocol (SMTP) [RFC5321], Extensible Messaging and Presence Protocol (XMPP) [RFC3920], XMPP Instant Messaging and Presence [RFC3921], vCard MIME Directory Profile [RFC2426], Internet Calendaring and Scheduling Core Object Specification (iCalendar) [RFC5545], and Calendaring Extensions to WebDAV (CalDAV) [RFC4791].

ICOM is open for extensions with additional domain models to enable seamless integration with business processes and social networks: for example in process integration domain which includes Business Process Model and Notation [BPMN], Web Services Business Process Execution Language [WS-BPEL], WS-BPEL Extension for People [BPEL4People], and Web Services for Human Task [WS-HumanTask]; in social networking domain, which includes Friend of a Friend [FOAF], Semantically-Interlinked Online Communities [SIOC], Open Social [OpenSocial], and Facebook Platform Open Graph [OpenGraph]. The OASIS ICOM TC Wiki [ICOM Wiki] provides Non-Normative supplemental information, including overview, primer, extensions, use cases, and mappings to various standard and proprietary data models.

The integrated model can be the foundation for defining the application programming interfaces (API) for application developers to develop integrated collaboration applications to interoperate with collaboration services. A service provider interface (SPI) can be specified to support interchangeable and interoperable services that conform to the ICOM application framework. ICOM does not prescribe how applications or services conforming to its model implement, store, or transport the data for objects.
1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2 Normative References


1.3 Non-Normative References


[OpenGraph] Facebook Platform Open Graph Core Concepts, (http://developers.facebook.com/docs/coreconcepts/)


2 Modeling Language

2.1 Introduction

ICOM specifies a set of objects in a collaboration environment, in terms of class definitions and property definitions of the classes. Objects comprise the information structures in a common application framework. An ICOM information structure MAY be composed of information from multiple repositories or collaboration services.

Note: To offer closer interoperability with OASIS Content Management Interoperability Services, ICOM specification follows the class and property definitions grammar of CMIS specification [CMIS], which is a normative reference for ICOM specification. ICOM specification adapts the CMIS class and property definitions grammar to introduce mixed-in types, enumeration types, and other base types which are not part of the domain model of CMIS Version 1 specification.

Note: One objective of ICOM standard is to offer seamless interoperability among identity management, content management, and collaboration services. Scope and Subject classes, defined respectively in Section 3.2 Scope Branch and Section 3.3 Subject Branch, can represent objects in Identity Management domain (such as LDAP). Artifact classes defined in Section 3.4 Artifact Branch can represent the extensions of CMIS Folder and Document base types. The extension modules in Section 4 define specialized subclasses of artifact and folder in Artifact Branch to support collaboration activities.

Note: ICOM extends the CMIS base types in several ways. ICOM Relationship class defined in Section 3.6.21 can represent n-ary relationships whereas CMIS Relationship base type represents binary relationships. ICOM version control model defined in Section 4.3.1 adopts the CMIS version control model and extends it with the concept of representative copy.

ICOM application framework includes a core model and a set of extension modules. All objects in the framework must be instances of at least one class.

Each class is defined in the class definition grammar, which specifies a namespace attribute, a localName attribute, a description attribute, an extendsFrom attribute representing a set of zero or more super classes, a stereotype attribute indicating whether a class is primary or mixin, an isAbstract attribute indicating whether a primary class is abstract, an isEnumeration attribute indicating whether instances of a primary class are enumerated, and a propertyDefinition attribute defining a set of zero or more properties of objects of the class. The properties are defined in the property definition grammar.

Note: The class and property definitions grammar corresponds to the UML meta-model, which is an OMG Meta Object Facility (MOF) M2-model. Each of the classes and properties thus defined are faithfully depicted by UML 2.0 diagrams in this specification.

A fully expanded class name, namespace/localName, MUST be unique within a domain.

Note: A namespace IRI reference qualifies a local name by associating the local name with the IRI reference to derive an expanded name.

2.2 Class Definition Grammar

A class-definition MUST contain the following attributes:

```
namespace String
```

The namespace attribute specifies an IRI.

```
localName String
```

The localName attribute specifies a local name portion of an expanded name or qualified name.
The description attribute describes the nature and intended use of a class.

The extendFrom attribute specifies a set of zero or more super classes.

The stereotype attribute specifies whether a class is a primary or mixin class.

A particular class is either a primary class or a mixin class, i.e. it cannot be both.

Inheritance is constrained by:

- a primary class MUST extend from one and only one primary class;
- a primary or mixin class MAY extend from zero or more mixin classes;
- a mixin class MUST NOT extend from a primary class.

An object MUST be an instance of one and only one primary class.

Note: When there is more than one super class in a class definition, at most one of the super classes is a primary class and the rest of the super classes are mixin classes. For example, Scope extends from Entity, RelationshipBondable, and Extent. Scope is a primary class. Among its super classes, only Entity is a primary class while RelationshipBondable and Extent are mixin classes.

The isAbstract attribute specifies whether a primary class is an abstract class. It is applicable only when the value of stereotype attribute is Primary.

The values of isAbstract attribute are:

- TRUE if the primary class is an abstract class;
- FALSE if the primary class is not an abstract class.

The default value is FALSE.

Note: An abstract class typically does not provide a complete declaration and cannot be instantiated. An abstract class is intended to be extended by other primary classes.

An abstract primary class MUST NOT extend from any non-abstract primary class.

The isEnumeration attribute specifies whether instances of a primary class are enumerated in a class definition. It is applicable only when the value of stereotype attribute is Primary.

The values of isEnumeration attribute are:

- TRUE if the instances of a primary class are enumerated in a class definition;
- FALSE if the instances of a primary class are not enumerated in a class definition.

The default value is FALSE.

Note: A primary class which is an enumeration of instances is also known as an enum class.
The `instances` attribute enumerates instances of an enum class. It is applicable only when the value of `stereotype` attribute is `Primary` and the value of `isEnumeration` attribute is `TRUE`.

The `propertyDefinition` attribute defines a set of zero or more property definitions for a class. Property definitions of a class are a union of inherited property definitions from super classes and property definitions explicitly defined on a class. The order of property definitions within a class is not significant. Property definitions MUST be uniquely named to avoid conflicts from multiple inheritances. Note: It is possible for the same property definition to be inherited through different paths in a super class hierarchy. Duplicate property definitions are eliminated from the set of property definitions of a class.

### 2.3 Property Definition Grammar

A `property-definition` MUST contain the following attributes:

- `namespace` String
  - The `namespace` attribute specifies an IRI.

- `localName` String
  - The `localName` attribute specifies the local name portion of an expanded name or qualified name.

- `description` String (optional)
  - The `description` attribute specifies a description of a property

- `propertyType` Enum
  - The `propertyType` attribute specifies a `property-type` for property values.
  - The value of `propertyType` attribute is one of the `property-type` names. The `property-type` names include names for the following data type defined by XML Schema Part 2 [XML SCHEMA]:
    - `string` (xsd:string)
    - `boolean` (xsd:boolean)
    - `decimal` (xsd:decimal)
    - `integer` (xsd:integer)
    - `datetime` (xsd:dateTime)
    - `duration` (xsd:duration)
    - `iri` (xsd:anyURI)
  - In addition, the following data type names are also specified by ICOM:
    - `id` (an opaque string representing an object id of an identifiable object)
    - `html` (a document or fragment of Hypertext Markup Language)
cardinality Enum

The cardinality attribute specifies a cardinality of property values.

The values of cardinality attribute are:

- **Single**: Property can have zero or one value (if property is not required), or exactly one value (if property is required)
- **Multi**: Property can have zero or more values (if property is not required), or one or more values (if property is required).

updatability Enum

The updatability attribute specifies under what circumstances the value of this property MAY be updated.

The values of updatability attribute are:

- **ReadOnly**: The value of this property MUST NOT be set directly by application. It is a property that is either maintained or computed by a service provider.
- **WriteOnly**: The value of this property can be set by application. It is a property whose value MAY be propagated into another ReadOnly property by a service provider.
- **ReadWrite**: The property value can be modified.
- **OnCreate**: The property value MUST only be update-able during the creation (a create operation) of an object.

inherited Boolean

The inherited attribute specifies whether a property definition is inherited from a super class.

The values of inherited attribute are:

- **TRUE** if a property definition is inherited from a super class;
- **FALSE** if a property definition is explicitly defined for a class.

required Boolean

The required attribute is only applicable to read-write and on-create properties, i.e. properties whose value is provided by application.

The values of required attribute are:

- **TRUE** if the value of a property MUST never be set to the “not set” state when an object of this type is created or updated. If a value is not provided during a create or update operation, a service provider MUST provide a value for the property. If a value is not provided, then a default value defined for the property MUST be set. If no default value is defined, a service provider MUST throw an exception.
- **FALSE** if the value of a property MAY be set to the “not set” state when an object of this type is created or updated.

This attribute is not applicable when the value updatability attribute is ReadOnly. In that case, required attribute SHOULD be set to FALSE.

Note: The value of a read-only property (such as icom_core:objectId, icom_core:createdBy) is set by a service provider. Hence, the value of the required attribute SHOULD be FALSE because it is read only for applications.
choices  property-choice-type (multi-valued)

The choices attribute specifies a set of single values allowed for this property.

Each value of choices attribute is an instance of property-choice-type that specifies a display name and a value to be stored in a property when selected.

If the value of cardinality attribute is Single and the value of openChoice attribute is FALSE, then a property value MUST be at most one of the values listed in choices attribute.

If the value of cardinality attribute is Single and the value of openChoice attribute is TRUE, then a property value MAY be one of the values listed in choices attribute.

If the value of cardinality attribute is Multi and the value of openChoice attribute is FALSE, then a property value MUST be zero, one, or more than one of the values listed in choices attribute.

If the value of cardinality attribute is Multi and the value of openChoice attribute is TRUE, then a property value MAY be zero, one, or more than one of the values listed in choices attribute.

If choices attribute is “not set”, then a property value MAY be an instance of the property-type specified by the propertyType attribute of a property definition.

openChoice  Boolean

The openChoice attribute specifies whether the value of a property must be listed in choices attribute. It is applicable only when choices attribute is set.

The values of openChoice attribute are:

- TRUE if a value of a property MAY be other than those listed in choices attribute;
- FALSE if a value of a property MUST be among those listed in choices attribute.

defaultValue  property-type

The defaultValue attribute specifies a value that a service provider MUST set for a property if a value is not provided by application when an object is created.

If no default value is specified and application creates an object of this class without setting a value for a property of this property definition, a service provider MUST attempt to store a “not set” state for the property value. If this occurs for a property that is defined to be required, then a service provider MUST throw an exception.

The value of the defaultValue attribute is an instance of the property-type specified by the propertyType attribute of a property definition.

minValue  Integer | Decimal

The minimum value allowed for a property. It is applicable only when the propertyType attribute of a property definition specifies the property types Integer or Decimal.

maxValue  Integer | Decimal

The maximum value allowed for a property. It is applicable only when the propertyType attribute of a property definition specifies the property types Integer or Decimal.
A property-choice-type MUST contain the following attributes:

- **displayName** String
  The `displayName` attribute specifies a string for presentation by application.

- **value** property-type
  The `value` attribute specifies a value compatible with the **property-type** specified by the `propertyType` attribute of a property definition.

### 2.4 Namespaces

Qualified names are subject to namespace interpretation depending on the namespace prefixes. A class definition includes the two attributes: namespace and localName. The namespace specifies one of the namespace prefixes in Table 1. The `localName` specifies an unprefixed name of a class. Syntactically, the namespace qualifies the local name.

#### Table 1 Namespace prefixes and IRI references.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>IRI Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core</td>
<td><a href="http://docs.oasis-open.org/ns/icom/core/201008">http://docs.oasis-open.org/ns/icom/core/201008</a></td>
</tr>
<tr>
<td>icom_ac</td>
<td><a href="http://docs.oasis-open.org/ns/icom/accesscontrol/201008">http://docs.oasis-open.org/ns/icom/accesscontrol/201008</a></td>
</tr>
<tr>
<td>icom_meta</td>
<td><a href="http://docs.oasis-open.org/ns/icom/metadata/201008">http://docs.oasis-open.org/ns/icom/metadata/201008</a></td>
</tr>
<tr>
<td>icom_content</td>
<td><a href="http://docs.oasis-open.org/ns/icom/content/201008">http://docs.oasis-open.org/ns/icom/content/201008</a></td>
</tr>
<tr>
<td>icom_doc</td>
<td><a href="http://docs.oasis-open.org/ns/icom/document/201008">http://docs.oasis-open.org/ns/icom/document/201008</a></td>
</tr>
<tr>
<td>icom_msg</td>
<td><a href="http://docs.oasis-open.org/ns/icom/message/201008">http://docs.oasis-open.org/ns/icom/message/201008</a></td>
</tr>
<tr>
<td>icom_card</td>
<td><a href="http://docs.oasis-open.org/ns/icom/contact/201008">http://docs.oasis-open.org/ns/icom/contact/201008</a></td>
</tr>
<tr>
<td>icom_presence</td>
<td><a href="http://docs.oasis-open.org/ns/icom/presence/201008">http://docs.oasis-open.org/ns/icom/presence/201008</a></td>
</tr>
<tr>
<td>icom_cal</td>
<td><a href="http://docs.oasis-open.org/ns/icom/calendar/201008">http://docs.oasis-open.org/ns/icom/calendar/201008</a></td>
</tr>
<tr>
<td>icom_task</td>
<td><a href="http://docs.oasis-open.org/ns/icom/task/201008">http://docs.oasis-open.org/ns/icom/task/201008</a></td>
</tr>
<tr>
<td>icom_forum</td>
<td><a href="http://docs.oasis-open.org/ns/icom/forum/201008">http://docs.oasis-open.org/ns/icom/forum/201008</a></td>
</tr>
<tr>
<td>icom_conf</td>
<td><a href="http://docs.oasis-open.org/ns/icom/conference/201008">http://docs.oasis-open.org/ns/icom/conference/201008</a></td>
</tr>
</tbody>
</table>

Note: The namespace prefix `icom_core` represents the IRI reference http://docs.oasis-open.org/ns/icom/core/201008 for ICOM core namespace. Both the unprefixed name Entity and prefixed name `icom_core:Entity` are qualified names that SHALL be interpreted by the expanded name http://docs.oasis-open.org/ns/icom/core/201008#Entity.
3 Core Model

3.1 Main Branch

3.1.1 Entity and Top-Level Subclasses

Figure 1 depicts Entity and top-level abstract classes forming the main branch of the ICOM class hierarchy. It depicts the Scope, Subject, and Artifact classes that represent the roots of the three major sub-branches of ICOM class hierarchy.

3.1.2 Identifiable

3.1.2.1 Description

An identifiable object has objectID and changeToken properties. The assignment of an objectID is implementation-dependent. The objectID is read only (immutable) once it is assigned.

3.1.2.2 Class Definition

The Identifiable class is a mixin class which defines the characteristics of entities and non-entities that enables unique identification.

The Identifiable class has attribute values:

```
localNamespace
  Value: icom_core

localName
  Value: Identifiable
```
373     extendsFrom
374     Value:
375
377     stereotype
378     Value: mixin
379
380     description
381     Value: Identifiable is a mixin class which defines the characteristics of all entities and some non-
382     entities that enables unique identification.
383
384     propertyDefinitions
385     The values for this attribute are defined in Section 3.1.2.3.
386
3.1.2.3 Property Definitions
387     The Identifiable class MUST have the property definitions:
388
389     icom_core:objectId
390     Description: A persistent identifier of an object.
391     Required: False
392     Inherited: False
393     Property Type: String
394     Cardinality: Single
395     Updatability: Read Only
396
397     icom_core:changeToken
398     Description: An opaque token used for optimistic locking & concurrency checking.
399     Required: False
400     Inherited: False
401     Property Type: String
402     Cardinality: Single
403     Updatability: Read Only
404
405     The Identifiable class MAY include additional property definitions which are implementation-defined.
406
3.1.3 Parental
3.1.3.1 Description
409     A parental object may be a parent of other objects.
3.1.3.2 Class Definition

The Parental class is a mixin class which defines the characteristics of entities that may be parents of other entities or identifiable objects.

The Parental class has attribute values:

- `localNamespace`
  - Value: icom_core

- `localName`
  - Value: Parental

- `extendsFrom`
  - Value: icom_core:Identifiable

- `stereotype`
  - Value: mixin

- `description`
  - Value: Parental is a mixin class which defines the characteristics of the entities that can be parents of other entities or identifiable objects.

`propertyDefinitions`

The values for this attribute are defined in Section 3.1.3.3.

3.1.3.3 Property Definitions

The Parental class inherits property definitions from super classes.

The Parental class MUST have the property definition:

- `icom_core:parent`
  - Description: Parent of an object.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Parental
  - Cardinality: Single
  - Updatability: Read Only

The Parental class MAY include additional property definitions which are implementation-defined.

3.1.4 Extent

3.1.4.1 Description

An extent object is a parental object which may contain other entities.
### 3.1.4.2 Class Definition

The Extent class is a mixin class which defines characteristics of entities that may contain other entities.

The Extent class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Extent

- **extendsFrom**
  - Value: icom_core:Parental

- **stereotype**
  - Value: mixin

- **description**
  - Value: Extent is a mixin class which defines the characteristics of entities that may contain other entities.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.1.4.3.

### 3.1.4.3 Property Definitions

The Extent class inherits property definitions from super classes.

The Extent class MUST have the property definition:

- **icom_core:parent**
  - Description: Parent of an extent.
  - Required: False
  - Inherited: True
  - Property Type: icom_core:Extent
  - Cardinality: Single
  - Updatability: Read Only

The Extent class MAY include additional property definitions which are implementation-defined.

### 3.1.5 Entity

#### 3.1.5.1 Description

An entity is an identifiable object that can be persisted and that has an access control list.
Each entity is assigned an internationalized resource identifier (IRI) composed from its objectld. The form of the IRI is implementation-dependent.

### 3.1.5.2 Class Definition

The Entity class has attribute values:

- `localNamespace`
  - Value: `icom_core`

- `localName`
  - Value: `Entity`

- `extendsFrom`
  - Value: `icom_core:Identifiable`

- `stereotype`
  - Value: `primary`

- `isAbstract`
  - Value: `TRUE`

- `description`
  - Value: An entity is an object with an immutable id and individual access control.

- `propertyDefinitions`
  - The values for this attribute are defined in Section 3.1.5.3.

### 3.1.5.3 Property Definitions

The Entity class inherits property definitions from super classes.

The Entity class MUST have the property definitions:

- `icom_core:name`
  - Description: Name of an entity.
  - Required: False
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write

- `icom_core:createdBy`
  - Description: An actor who created an entity.
  - Required: False
  - Inherited: False
Property Type: icom_core:Actor
Cardinality: Single
Updatability: Read Only

icom_core:creationDate
Description: Date and time when an entity is created. It is immutable.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Only

icom_core:lastModifiedBy
Description: An actor who last modified an entity.
Required: False
Inherited: False
Property Type: icom_core:Actor
Cardinality: Single
Updatability: Read Only

icom_core:lastModificationDate
Description: Date and time of last modification.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Only

icom_core:parent
Description: A parental entity which contains an entity.
Required: False
Inherited: False
Property Type: icom_core:Parental
Cardinality: Single
Updatability: Read Only

icom_ac:owner
Description: A subject who owns an entity.
Required: True
Inherited: False
Property Type: icom_ac:Owner
Cardinality: Single
Updatability: Read Write

icom_ac:accessControlList
Description: Access control list on an entity.
Required: False
Inherited: False
Property Type: icom_ac:AccessControlList
Cardinality: Single
Updatability: Read Write

icom_meta:attachedMarker
Description: Zero or more markers applied on an entity.
Required: False
Inherited: False
Property Type: icom_meta:Marker
Cardinality: Multi
Updatability: Read Only

icom_meta:categoryApplication
Description: Zero or more category applications on an entity.
Required: False
Inherited: False
Property Type: icom_meta:CategoryApplication
Cardinality: Multi
Updatability: Read Only

icom_meta:tagApplication
Description: Zero or more tag applications on an entity.
Required: False
Inherited: False
Property Type: icom_meta:TagApplication
Cardinality: Multi
Updatability: Read Only

The Entity class MAY include additional property definitions which are implementation-defined.
3.1.6 Overview of Scope, Subject, and Artifact Branches

The UML diagram in Figure 3 depicts the core classes in the Scope, Subject, and Artifact branches of ICOM class hierarchy. Scope branch includes the model of communities and spaces which are containers of subjects and artifacts. Subject branch includes the model of actors, groups, and roles. Artifact branch includes the model of content and metadata produced by actors.

Note: The Subject and Artifact branches support the separation of concerns of user administration and content management. Typically subjects and artifacts are joined in the (subject, privilege, artifact) triples of access control model. Some of the (subject, privilege, artifact) triples are derived from the scopes of the role assignments and the artifacts contained by the scopes. The communities and spaces contain subjects and artifacts; however, membership of subjects in a space is administered separately from management of artifacts in the space.

Scope, Subject, and Artifact are defined in Section 3.2, 3.3, and 3.4, respectively.
3.2 Scope Branch

3.2.1 Scope and Top-Level Subclasses

Figure 4: Scope Branch.

Figure 4 depicts the top-level classes of Scope Branch, which includes Scope, Community, and Space.
3.2.2 Scope

3.2.2.1 Description

A scope is an extent of an administrative policy.

3.2.2.2 Class Definition

The Scope class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Scope

- **extendsFrom**
  - Value: icom_core:Entity, icom_core:Extent, icom_meta:RelationshipBondable

- **stereotype**
  - Value: primary

- **isAbstract**
  - Value: TRUE

- **description**
  - Value: A scope is an extent of an administrative realm.

3.2.2.3 Property Definitions

The Scope class inherits property definitions from super classes.

The Scope class MUST have the property definitions:

- **icom_core:description**
  - Description: A description of a scope.
  - Required: False
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:parent**
  - Description: A community which contains a scope.
<table>
<thead>
<tr>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>True</td>
<td>icom_core:Community</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>icom_core:group</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_core:memberGroup</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_ac:roleDefinition</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_ac:role</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_meta:relationship</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
</tbody>
</table>
The Scope class MAY include additional property definitions which are implementation-defined.

Figure 5: Scope Class Diagram.

3.2.3 Community

3.2.3.1 Description

A community is a scope that has a set of actors as members who can participate in a set of spaces. It is implementation-dependent whether or not a space in a community can include participating actors who are not members of a parent community or ancestor communities.

3.2.3.2 Class Definition

The Community class has attribute values:

- **localNamespace**
  - Value: `icom_core`

- **localName**
  - Value: `Community`
extendsFrom
    Value: icom_core:Scope

stereotype
    Value: primary

description
    Value: A community is a scope that has a set of actors as members who can participate in a set of spaces.

propertyDefinitions
    The values for this attribute are defined in Section 3.2.3.3.

### 3.2.3.3 Property Definitions

The Community class inherits property definitions from super classes.

The Community class MUST have the property definitions:

- **icom_core:community**
  - Description: Sub-communities of a community.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Community
  - Cardinality: Multi
  - Updatability: Read Only

- **icom_core:space**
  - Description: Spaces of a community.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Space
  - Cardinality: Multi
  - Updatability: Read Only

- **icom_core:actor**
  - Description: Managed actors of a community, i.e. actors whose parent community is this community.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Actor
  - Cardinality: Multi
  - Updatability: Read Only
icom_core:memberActor

Description: Member actors of a community, i.e. actors whose assigned communities include this community.

Required: False
Inherited: False
Property Type: icom_core.Actor
Cardinality: Multi
Updatability: Read Only

The Community class MAY include additional property definitions which are implementation-defined.

Figure 6: Community Class Diagram.

3.2.4 Space

3.2.4.1 Description
A space is a scope that defines a durable context and place for actors to work or collaborate.

3.2.4.2 Class Definition
The Space class has attribute values:

localNamespace
Value: icom_core

localName
Value: Space

extendsFrom
Value: icom_core:Scope, icom_core:FolderContainer
stereotype
Value: primary

description
Value: A space is a scope that defines a durable context and place for actors to work or collaborate.

propertyDefinitions
The values for this attribute are defined in Section 3.2.4.3.

3.2.4.3 Property Definitions
The Space class inherits property definitions from super classes.
The Space class MUST have the property definition:

icom_core:element
Description: Elements of a space.
Required: False
Inherited: True
Property Type: icom_core:SpaceItem
Cardinality: Multi
Updatability: Read Only

The Space class MAY include additional property definitions which are implementation-defined.

Figure 7: Space Class Diagram.
3.3 Subject Branch

3.3.1 Subject and Top-Level Subclasses

Figure 8: Subject Branch.

Figure 8 depicts the top-level classes of Subject Branch, which includes Subject, Role, Group, and Actor.

3.3.2 Subject

3.3.2.1 Description

A subject is an entity that can have rights to perform actions.

3.3.2.2 Class Definition

The Subject class has attribute values:

- `localNamespace`
  - Value: icom_core

- `localName`
  - Value: Subject

- `extendsFrom`
  - Value: icom_core:Entity, icom_meta:RelationshipBondable

- `stereotype`
  - Value: primary

- `isAbstract`
  - Value: TRUE

- `description`
  - Value: A subject is an entity that can have rights to perform actions.
The values for this attribute are defined in Section 3.3.2.3.

### 3.3.2.3 Property Definitions

The Subject class inherits property definitions from super classes. The Subject class MUST have the property definitions:

**icom_core:description**
- Description: A description of a subject.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:parent**
- Description: A scope which contains a subject.
- Required: False
- Inherited: True
- Property Type: icom_core:Scope
- Cardinality: Single
- Updatability: Read Only

**icom_meta:relationship**
- Description: Zero or more relationships associated with a subject.
- Required: False
- Inherited: False
- Property Type: icom_meta:Relationship
- Cardinality: Multi
- Updatability: Read Only

**icom_meta:property**
- Description: Zero or more extended properties of a subject.
- Required: False
- Inherited: False
- Property Type: icom_meta:Property
- Cardinality: Multi
- Updatability: Read Write

The Subject class MAY include additional property definitions which are implementation-defined.
3.3.3 Group

3.3.3.1 Description
A group is a subject representing a set of actors and sub-groups.
A group can be part of one or more super-groups.
It can be an owner of one or more entities.

3.3.3.2 Class Definition
The Group class has attribute values:

- **localNamespace**
  Value: icom_core

- **localName**
  Value: Group

- **extendsFrom**
  Value: icom_core:Subject, icom_core:Addressable, icom_ac:Accessor
  Optional Value: icom_ac:Owner

- **stereotype**
  Value: primary

- **description**
  Value: A group is a subject representing a set of actors and sub-groups. A group can be part of one or more super-groups. It can be an owner of one or more entities.

- **propertyDefinitions**
  The values for this attribute are defined in Section 3.3.3.3.
3.3.3.3 Property Definitions

The Group class inherits property definitions from super classes. The Group class MUST have the property definitions:

icom_core:assignedGroup
  Description: A group's super-groups.
  Required: False
  Inherited: False
  Property Type: icom_core:Group
  Cardinality: Multi
  Updatability: Read Write

icom_core:assignedScope
  Description: A group's scopes.
  Required: False
  Inherited: False
  Property Type: icom_core:Scope
  Cardinality: Multi
  Updatability: Read Write

icom_core:memberGroup
  Description: Sub-groups of a group.
  Required: False
  Inherited: False
  Property Type: icom_core:Group
  Cardinality: Multi
  Updatability: Read Only

icom_core:memberActor
  Description: Actors in a group.
  Required: False
  Inherited: False
  Property Type: icom_core:Actor
  Cardinality: Multi
  Updatability: Read Only

icom_ac:assignedRole
  Description: A group's roles.
  Required: False
  Inherited: False
  Property Type: icom_ac:Role
  Cardinality: Multi
The Group class MAY include additional property definitions which are implementation-defined.

Figure 10: Group and Actor Class Diagram.

3.3.4 Actor

3.3.4.1 Description
An actor is a subject that can perform actions on objects.
It can be an owner of entities.

3.3.4.2 Class Definition
The Actor class has attribute values:

- `localNamespace`
  - Value: icom_core

- `localName`
  - Value: Actor
extendsFrom
  Value: icom_core:Subject, icom_core:Addressable, icom_ac:Owner

stereotype
  Value: primary

isAbstract
  Value: TRUE

description
  Value: An actor is a subject that can perform actions on objects.

propertyDefinitions
  The values for this attribute are defined in Section 3.3.4.3.

### 3.3.4.3 Property Definitions

The Actor class inherits property definitions from super classes.

The Actor class MUST have the property definitions:

**icom_core:parent**

Description: A community which contains an actor.

Required: False

Inherited: True

Property Type: icom_core:Community

Cardinality: Single

Updatability: Read Only

**icom_core:assignedGroup**

Description: An actor’s groups.

Required: False

Inherited: False

Property Type: icom_core:Group

Cardinality: Multi

Updatability: Read Write

**icom_core:assignedCommunity**

Description: An actor’s communities.

Required: False

Inherited: False

Property Type: icom_core:Community

Cardinality: Multi

Updatability: Read Write
3.3.5 Person

3.3.5.1 Description
A person is an individual human who may be an actor.
A person has a personal space.

3.3.5.2 Class Definition
The Person class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Person

- **extendsFrom**
  - Value: icom_core:Actor

- **stereotype**
  - Value: primary

- **description**
  - Value: A person is an individual human who may be an actor.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.3.5.3.

3.3.5.3 Property Definitions
The Person class inherits property definitions from super classes.
The Person class MUST have the property definitions:

**icom_core:givenName**
- Description: Given name of a person.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:middleName**
- Description: Middle name of a person. Can include multiple names concatenated.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:familyName**
- Description: Family name of a person.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:prefix**
- Description: Prefix of a person’s name.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:suffix**
- Description: Suffix of a person’s name.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write
icom_core:nickname
Description: Nickname of a person.
Required: False
Inherited: False
Property Type: String
Cardinality: Multi
Updatability: Read Write

icom_core:jobTitle
Description: Job title of a person.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:department
Description: A person’s affiliated department.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:officeLocation
Description: Location of a person’s department.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:company
Description: A person’s affiliated company.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:profession
Description: A person’s profession.
The Person class MAY include additional property definitions which are implementation-defined.
3.3.6 Resource

3.3.6.1 Description

A resource is an actor representing a bookable resource, such as a conference room, equipment, or on-line conference.

A resource is associated with a resource space that contains a resource scheduling calendar and on-line conference.

3.3.6.2 Class Definition

The Resource class has attribute values:

```plaintext
localNamespace
Value: icom_core

describeValue
Value: Resource

eextendFrom
Value: icom_core:Actor

describeType
Value: primary

describeValue
Value: A resource actor is an actor representing a bookable resource, such as a conference room, equipment, or on-line conference.

propertyDefinitions
The values for this attribute are defined in Section 3.3.6.3.
```
3.3.6.3 Property Definitions

The Resource class inherits property definitions from super classes. The Resource class MUST have the property definitions:

icom_core:resourceSpace
- Description: Administrative space of a resource actor.
- Required: False
- Inherited: False
- Property Type: icom_core:Space
- Cardinality: Single
- Updatability: Read Only

icom_core:location
- Description: Location of a resource.
- Required: False
- Inherited: False
- Property Type: icom_core:Location
- Cardinality: Single
- Updatability: Read Write

icom_core:capacity
- Description: Capacity of a resource.
- Required: False
- Inherited: False
- Property Type: Integer
- Cardinality: Single
- Updatability: Read Write

icom_core:resourceType
- Description: Type of a resource.
- Required: False
- Inherited: False
- Property Type: icom_core:ResourceType
- Cardinality: Single
- Updatability: Read Write

icom_core:bookingRule
- Description: Resource booking rule.
- Required: False
- Inherited: False
- Property Type: icom_core:ResourceBookingRule
- Cardinality: Single
**Updatability:**
Read Write

**icom_core:bookingApprover**

- **Description:** One or more persons who approve the booking of a resource.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:Person
- **Cardinality:** Multi
- **Updatability:** Read Write

The Resource class MAY include additional property definitions which are implementation-defined.

**Figure 12: Resource Class Diagram.**

### 3.3.7 ResourceType

#### 3.3.7.1 Description

A resource type is a category of resources.

#### 3.3.7.2 Class Definition

The ResourceType class is a mixin class which defines a resource type.

The ResourceType class has attribute values:

- **localNamespace**
  - Value: icom_core
localName
  Value: ResourceType

extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: ResourceType is a mixin class which defines a type of resources.

propertyDefinitions
  The values for this attribute are defined in Section 3.3.7.3.

3.3.7.3 Property Definitions
The ResourceType class MAY include additional property definitions which are implementation-defined.

3.3.8 ResourceTypeEnum
The ResourceTypeEnum class is an enum class that enumerates the instances each of which expresses a type of resources.
The ResourceTypeEnum class has attribute values:

localNamespace
  Value: icom_core

localName
  Value: ResourceTypeEnum

extendsFrom
  Value: icom_core:ResourceType

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: A type of resources.
ICOM defines four resource types:

- **icom_core:Room** a resource represents a room.
- **icom_core:Equipment** a resource represents an equipment.
- **icom_core:OnlineConference** a resource represents an online conference.
- **icom_core:OtherResourceType** a resource represents other things.

### 3.3.9 ResourceBookingRule

#### 3.3.9.1 Description

A resource booking rule is a strategy for allocating resources for calendar scheduling.

#### 3.3.9.2 Class Definition

The ResourceBookingRule class is a mixin class which defines a resource booking rule.

The ResourceBookingRule class has attribute values:

- **localNamespace**
  - Value: icom_core
- **localName**
  - Value: ResourceBookingRule
- **extendsFrom**
  - Value: 
- **stereotype**
  - Value: mixin
- **description**
  - Value: ResourceBookingRule is a mixin class which defines a rule for allocating resources for calendar scheduling.
- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.3.9.3.

### 3.3.9.3 Property Definitions

The ResourceBookingRule class MAY include additional property definitions which are implementation-defined.
3.3.10 ResourceBookingRuleEnum

The ResourceBookingRuleEnum class is an enum class that enumerates the instances each of which expresses a booking rule.

The ResourceBookingRuleEnum class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: ResourceBookingRuleEnum

- **extendsFrom**
  - Value: icom_core:ResourceBookingRule

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A resource booking rule for allocating resources for calendar scheduling.

- **instances**
  - Value: <icom_core:Open, icom_core:FirstComeFirstServed>

ICOM defines two resource booking rules:

- **icom_core:Open** a resource is open for booking.
- **icom_core:FirstComeFirstServed** a resource is first come first served.
3.4 Artifact Branch

3.4.1 Artifact and Top-Level Subclasses

Figure 13: Artifact Branch.

Figure 13 depicts the top-level classes of Artifact Branch, which includes Artifact, Folder, HeterogeneousFolder, Document, and Message.

3.4.2 Item

3.4.2.1 Description

An item is an element of a container.
The parent of an item MUST be a container.

3.4.2.2 Class Definition

The Item class is a mixin class which defines the characteristics of entities that can be elements of a Container.
The Item class has attribute values:
localNamespace
  Value: icom_core

localName
  Value: Item

extendsFrom
  Value: icom_core:Identifiable

stereotype
  Value: mixin

description
  Value: Item is a mixin class which defines the characteristics of entities that can be placed in a Container.

propertyDefinitions
  The values for this attribute are defined in Section 3.4.2.3.

3.4.2.3 Property Definitions
The Item class inherits property definitions from super classes.
The Item class MUST have the property definition:

icom_core:parent
  Description: A parent container of an item.
  Required: False
  Inherited: True
  Property Type: icom_core:Container
  Cardinality: Single
  Updatability: Read Only

The Item class MAY have the optional property definition:

icom_core:container
  Description: Zero, one, or more containers of an item, including the parent container.
  Required: False
  Inherited: False
  Property Type: icom_core:Container
  Cardinality: Multi
  Updatability: Read Write

The Item class MAY include additional property definitions which are implementation-defined.
3.4.3 Spaceltem

3.4.3.1 Description
A space item is an item that can be an element of a space.

3.4.3.2 Class Definition
The Spaceltem class is a mixin class which defines the characteristics of items that can be elements of a Space.

The Spaceltem class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Spaceltem

- **extendsFrom**
  - Value: icom_core:Item

- **stereotype**
  - Value: mixin

- **description**
  - Value: Spaceltem is a mixin class which defines the characteristics of entities that can be elements of a Space.

3.4.3.3 Property Definitions
The Spaceltem class inherits property definitions from super classes.
The Spaceltem class MAY include additional property definitions which are implementation-defined.

3.4.4 Container

3.4.4.1 Description
A container is an extent that contains items.

3.4.4.2 Class Definition
The Container class is a mixin class which defines the characteristics of extents that contain items.
The Container class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Container

- **extendsFrom**
  - Value: icom_core:Extent

- **stereotype**
  - Value: mixin

- **description**
  - Value: A container is an extent that contains items.

**3.4.4.3 Property Definitions**

The Container class inherits property definitions from super classes.

The Container class MUST have the property definition:

- **icom_core:element**
  - Description: Elements of a container, i.e. items whose parent container is the container or whose containers include the container.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Item
  - Cardinality: Multi
  - Updatability: Read Only

The Container class MAY include additional property definitions which are implementation-defined.

**3.4.5 FolderContainer**

**3.4.5.1 Description**

A folder container is a container which may contain folders. Space and heterogeneous folder are folder containers.
3.4.5.2 Class Definition

The FolderContainer class is a mixin class that defines the characteristics of containers that may contain folders.

The FolderContainer class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: FolderContainer

extendsFrom
  Value: icom_core:Container

stereotype
  Value: mixin

description
  Value: A folder container is a container which may contain folders.

propertyDefinitions
  The values for this attribute are defined in Section 3.4.5.3.
```

3.4.5.3 Property Definitions

The FolderContainer class inherits property definitions from super classes.

The FolderContainer class MAY include additional property definitions which are implementation-defined.

3.4.6 Artifact

3.4.6.1 Description

An artifact is a result of a communication, cooperation, content creation, or collaboration activity.

Note: Document versioning is an example of content creation activity resulting in an artifact (a version of a document).

3.4.6.2 Class Definition

The Artifact class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Artifact
```
extendsFrom
   Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
   Optional Value: icom_core:SpaceItem

stereotype
   Value: primary

isAbstract
   Value: TRUE

description
   Value: An artifact is a result of a communication, cooperation, content creation, or collaboration activity.

propertyDefinitions
   The values for this attribute are defined in Section 3.4.6.3.

3.4.6.3 Property Definitions
The Artifact class inherits property definitions from super classes.
The Artifact class MUST have the property definitions:

icom_core:description
   Description: A description of an artifact.
   Required: False
   Inherited: False
   Property Type: String
   Cardinality: Single
   Updatability: Read Write

icom_core:userCreationDate
   Description: Date and time when an artifact was created.
   Required: False
   Inherited: False
   Property Type: DateTime
   Cardinality: Single
   Updatability: Read Write

icom_core:userLastModificationDate
   Description: Date and time when an artifact was last modified.
   Required: False
   Inherited: False
   Property Type: DateTime
   Cardinality: Single
Updatability: Read Write

icom_meta:property
Description: Zero or more extended properties of an artifact.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

icom_meta:viewerProperty
Description: Zero or more extended properties of an artifact visible to a viewer.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

icom_meta:relationship
Description: Zero or more relationships associated with an artifact.
Required: False
Inherited: False
Property Type: icom_meta:Relationship
Cardinality: Multi
Updatability: Read Only

The Artifact class MAY include additional property definitions which are implementation-defined.
### 3.4.7 Folder

#### 3.4.7.1 Description

A folder is an artifact that may contain other artifacts.

Note: Every folder except root folders has at least one parent folder. The parent of a root folder is a space. Subclasses of Folder class should enforce their own semantics on elements.

#### 3.4.7.2 Class Definition

The Folder class has attribute values:

- `localNamespace`
  - Value: `icom_core`

- `localName`
  - Value: `Folder`

- `extendsFrom`
  - Value: `icom_core:Artifact, icom_core:Container, icom_core:SpaceItem`

- `stereotype`
  - Value: `primary`

- `isAbstract`
  - Value: `TRUE`
description
Value: A folder is an artifact that may contain other artifacts.

propertyDefinitions
The values for this attribute are defined in Section 3.4.7.3.

3.4.7.3 Property Definitions

The Folder class inherits property definitions from super classes.

The Folder class MUST have the property definition:

icom_core:parent
Description: A parent container of a folder.
Required: False
Inherited: True
Property Type: icom_core:FolderContainer
Cardinality: Single
Updatability: Read Only

The Folder class MAY include additional property definitions which are implementation-defined.

3.4.8 HeterogeneousFolder

3.4.8.1 Description
A heterogeneous folder is an unconstrained folder to contain any type of artifacts.

Note: It is typically used for document folders, inbox, outbox, and trash folder of a space.

3.4.8.2 Class Definition

The HeterogeneousFolder class has attribute values:

localNamespace
Value: icom_core

localName
Value: HeterogeneousFolder

extendsFrom
Value: icom_core:Folder, icom_core:FolderContainer

stereotype
Value: primary
description
Value: A heterogeneous folder is an unconstrained folder to contain any type of artifacts.

propertyDefinitions
The values for this attribute are defined in Section 3.4.8.3.

3.4.8.3 Property Definitions
The HeterogeneousFolder class inherits property definitions from super classes.
The HeterogeneousFolder class MUST have the property definition:

icom_core:element
Description: Elements of a heterogeneous folder.
Required: False
Inherited: True
Property Type: icom_core:Artifact
Cardinality: Multi
Updatability: Read Only

The HeterogeneousFolder class MAY include additional property definitions which are implementation-defined.

Figure 15: Heterogeneous Folder Class Diagram.
3.5 Access Control Model

3.5.1 Accessor

3.5.1.1 Description
An accessor can be granted or denied access rights to objects.

3.5.1.2 Class Definition
The Accessor class is a mixin class which defines the characteristics of subjects such as groups and actors that can be granted or denied access types in access control lists and privileges in role assignments.

The Accessor class has attribute values:

- **localNamespace**
  - Value: icom_ac

- **localName**
  - Value: Accessor

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: mixin

- **description**
  - Value: Accessor is a mixin class which defines the characteristics of subjects such as groups and actors that can be granted or denied access types in access control lists and granted privileges in role assignments.

3.5.1.3 Property Definitions
The Accessor class inherits property definitions from super classes.

The Accessor class MAY include additional property definitions which are implementation-defined.

3.5.2 Owner

3.5.2.1 Description
An owner is a subject that can be the owner of entities.

An owner of an entity MAY always have rights to update the access control list for the entity.
3.5.2.2 Class Definition
The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors that can own entities.

The Owner class has attribute values:

- `localNamespace`
  - Value: icom_ac
- `localName`
  - Value: Owner
- `extendsFrom`
  - Value: icom_ac:Accessor
- `stereotype`
  - Value: mixin
- `description`
  - Value: Owner is a mixin class which defines the characteristics of subjects such as groups and actors that can own entities.

3.5.2.3 Property Definitions
The Owner class inherits property definitions from super classes.
The Owner class MAY include additional property definitions which are implementation-defined.

3.5.3 RoleDefinition
3.5.3.1 Description
A role definition is a named set of privileges.

3.5.3.2 Class Definition
The RoleDefinition class has attribute values:

- `localNamespace`
  - Value: icom_ac
- `localName`
  - Value: RoleDefinition
extendsFrom
Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype
Value: primary

description
Value: A role definition is a named set of privileges.

propertyDefinitions
The values for this attribute are defined in Section 3.5.3.3.

3.5.3.3 Property Definitions
The RoleDefinition class inherits property definitions from super classes.
The RoleDefinition class MUST have the property definition:

icom_core:description
Description: A description of a role definition.
 Required: False
 Inherited: False
 Property Type: String
 Cardinality: Single
 Updatability: Read Write

icom_ac:privilege
Description: A set of privileges.
 Required: True
 Inherited: False
 Property Type: icom_ac:Privilege
 Cardinality: Multi
 Updatability: Read Write

The RoleDefinition class MAY include additional property definitions which are implementation-defined.

3.5.4 Role

3.5.4.1 Description
A role assigns a named set of privileges to a set of accessors for operations within an assigned scope.

3.5.4.2 Class Definition
The Role class has attribute values:
localNamespace
  Value: icom_ac

localName
  Value: Role

extendsFrom
  Value: icom_core:Subject

stereotype
  Value: primary

description
  Value: A role assigns a named set of rights to a set of accessors for operations within an
  assigned scope.

propertyDefinitions
  The values for this attribute are defined in Section 3.5.4.3.

3.5.4.3 Property Definitions
The Role class inherits property definitions from super classes.
The Role class MUST have the property definitions:

icom_ac:roleDefinition
  Description: A role definition containing a set of privileges.
  Required: True
  Inherited: False
  Property Type: icom_ac:RoleDefinition
  Cardinality: Single
  Updatability: On Create

icom_ac:assignedScope
  Description: A scope in which a role is assigned.
  Required: True
  Inherited: False
  Property Type: icom_core:Scope
  Cardinality: Single
  Updatability: Read Write

icom_ac:memberAccessor
  Description: Accessors (actors and groups) assigned to a role.
  Required: False
  Inherited: False
The Role class MAY include additional property definitions which are implementation-defined.

Figure 16: Role Definition and Role Class Diagram.

3.5.5 Privilege

3.5.5.1 Description

A privilege is an access right granted through roles.

3.5.5.2 Class Definition

The Privilege class is a mixin class which defines access rights that can be included in role definitions.

The Privilege class has attribute values:

- **localNamespace**
  - Value: icom_ac

- **localName**
  - Value: Privilege
extendsFrom
    Value: 

stereotype
    Value: mixin

description
    Value: Privilege is a mixin class which defines access rights that can be included in role definitions.

propertyDefinitions
    The values for this attribute are defined in Section 3.5.5.3.

3.5.5.3 Property Definitions
The Privilege class MAY include additional property definitions which are implementation-defined.

3.5.6 PrivilegeEnum
The PrivilegeEnum class is an enum class that enumerates the instances each of which expresses a privilege that can be assigned to a role.
The PrivilegeEnum class has attribute values:

    localNamespace
    Value: icom_ac

    localName
    Value: PrivilegeEnum

    extendsFrom
    Value: icom_ac:Privilege

    stereotype
    Value: primary

    isEnumeration
    Value: TRUE

    description
    Value: Privilege that can be assigned to a role.

    instances
    Value: <icom_ac:Archive, icom_ac:Audit>
ICOM defines two privileges:
- `icom_ac:Archive` a right to archive contents in a scope.
- `icom_ac:Audit` a right to audit activities in a scope.

### 3.5.7 AccessControlList

#### 3.5.7.1 Description
An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the entity.

#### 3.5.7.2 Class Definition
The AccessControlList class has attribute values:

- **localNamespace**
  - Value: `icom_ac`

- **localName**
  - Value: `AccessControlList`

- **extendsFrom**
  - Value: 

- **stereotype**
  - Value: `primary`

- **description**
  - Value: An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the entity.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.5.7.3.

#### 3.5.7.3 Property Definitions
The AccessControlList class MUST have the property definitions:

- **icom_ac:object**
  - Description: Associated object.
  - Required: True
  - Inherited: False
  - Property Type: `icom_core:Entity`
  - Cardinality: Single
  - Updatability: On Create
**AccessControlEntry**

**3.5.8.1 Description**

An access control entry specifies access types granted to or denied for an accessor.

**3.5.8.2 Class Definition**

The AccessControlEntry class has attribute values:

- **localNamespace**
  - Value: icom_ac

- **localName**
  - Value: AccessControlEntry

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: An access control entry is associated with an accessor and contains a list of access types (permissions) granted to or denied from the accessor.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.5.8.3.

**3.5.8.3 Property Definitions**

The AccessControlEntry class MUST have the property definitions:

- **icom_ac:subject**
  - Description: Associated subject.
  - Required: True
3.5.9 AccessType

An AccessType is an access right granted through an access control entry.

3.5.9.1 Class Definition

The AccessType class is a mixin class which defines access rights that can be granted or denied in an access control entry.

The AccessType class has attribute values:

```plaintext
localNamespace

    Value: icom_ac

localName

    Value: AccessType

extendsFrom

    Value:
```
stereotype
    Value: mixin

description
    Value: AccessType is a mixin class which defines access rights that can be granted or denied in an access control entry.

propertyDefinitions
    The values for this attribute are defined in Section 3.5.9.2.

3.5.9.2 Property Definitions
The AccessType class inherits property definitions from super classes.
The AccessType class MAY include additional property definitions which are implementation-defined.

3.5.10 AccessTypeEnum
The AccessTypeEnum class is an enum class that enumerates the instances each of which expresses an access type that can be granted or denied in an access control entry.
The AccessTypeEnum class has attribute values:

localNamespace
    Value: icom_ac

localName
    Value: AccessTypeEnum

extendsFrom
    Value: icom_ac:AccessType

stereotype
    Value: primary

isEnumeration
    Value: TRUE

description
    Value: Access type that can be granted or denied in an access control entry.

instances
    Value: <icom_ac:Read, icom_ac:Write, icom_ac:Delete>

ICOM defines three access types:
- icom_ac:Read a right to retrieve an entity.
- **icom_ac:Write** a right to update an entity.
- **icom_ac:Delete** a right to delete an entity.

**Figure 17: Access Control List Class Diagram.**

### 3.6 Metadata Model

#### 3.6.1 ClassDefinition

#### 3.6.1.1 Description

A class definition is an entity that defines a type of entities.

#### 3.6.1.2 Class Definition

The `ClassDefinition` class has attribute values:

- **localNamespace**
  - Value: `icom_meta`

- **localName**
  - Value: `ClassDefinition`
2125   extendsFrom
2126       Value: icom_core:Entity, icom_meta:RelationshipBondable
2128
2129   stereotype
2130       Value: primary
2131
2132   isAbstract
2133       Value: FALSE
2134
2135   description
2136       Value: A class definition defines a type of entities.
2137
2138   propertyDefinitions
2139       The values for this attribute are defined in Section 3.6.1.3.

2140 3.6.1.3 Property Definitions
2141 The ClassDefinition class inherits property definitions from super classes.
2142 The ClassDefinition class MUST have the property definition:

2143 icom_core:namespace
2144       Description: Namespace for a class name.
2145       Required: False
2146       Inherited: False
2147       Property Type: String
2148       Cardinality: Single
2149       Updatability: Read Write
2150
2152 icom_core:description
2153       Description: A description of a class.
2154       Required: False
2155       Inherited: False
2156       Property Type: String
2157       Cardinality: Single
2158       Updatability: Read Write
2159
2160 icom_meta:extendsFrom
2161       Description: One or more generalizations of a class.
2162       Required: True
2163       Inherited: False
2164       Property Type: icom_meta:ClassDefinition
2165       Cardinality: Multi
2166       Updatability: Read Write
icom_meta:stereotype

- Description: Stereo type of a class.
- Required: True
- Inherited: False
- Property Type: icom_meta:Stereotype
- Cardinality: Single
- Updatability: Read Write

icom_meta:abstract

- Description: Indicates whether a class is abstract or concrete.
- Required: False
- Inherited: False
- Property Type: Boolean
- Cardinality: Single
- Updatability: Read Write

icom_meta:enumeration

- Description: Indicates whether instances of a class are enumerated. This property is applicable only if the stereotype property is primary.
- Required: False
- Inherited: False
- Property Type: Boolean
- Cardinality: Single
- Updatability: Read Write

icom_meta:instances

- Description: Instances of an enumeration class. This property is applicable only if the enumeration property is true.
- Required: False
- Inherited: False
- Property Type: IRI
- Cardinality: Multi
- Updatability: Read Write

icom_meta:propertyDefinition

- Description: One or more property definitions of a class definition.
- Required: False
- Inherited: False
- Property Type: icom_meta:PropertyDefinition
- Cardinality: Multi
- Updatability: Read Write
The ClassDefinition class MAY include additional property definitions which are implementation-defined.

Figure 18: Class Definition and Property Definition Class Diagram.

3.6.2 Stereotype

3.6.2.1 Description

A stereotype of a class definition.
3.6.2.2 Class Definition

The StereoType class is a mixin class which defines a stereo type of a class definition. The StereoType class has attribute values:

- `localNamespace`
  - Value: `icom_meta`

- `localName`
  - Value: `StereoType`

- `extendsFrom`
  - Value:

- `stereotype`
  - Value: `mixin`

- `description`
  - Value: StereoType is a mixin class which defines a stereo type of a class definition.

- `propertyDefinitions`
  - The values for this attribute are defined in Section 3.6.2.3.

3.6.2.3 Property Definitions

The StereoType class MAY include additional property definitions which are implementation-defined.

3.6.3 StereoTypeEnum

The StereoTypeEnum class is an enum class that enumerates the instances each of which defines a stereo type of a class definition. The StereoTypeEnum has attribute values:

- `localNamespace`
  - Value: `icom_meta`

- `localName`
  - Value: `StereoTypeEnum`

- `extendsFrom`
  - Value: `icom_meta:StereoType`

- `stereotype`
  - Value: `primary`
isEnumeration

Value: TRUE

description

Value: Stereo type of a class definition.

instances

Value: <icom_meta:Primary, icom_meta:Mixin>

ICOM defines two stereo types:

- icom_meta:Primary a primary class.
- icom_meta:Mixin a mixin class.

3.6.4 PropertyDefinition

3.6.4.1 Description

A property definition specifies the name, type, choice, and cardinality of values for properties.

3.6.4.2 Class Definition

The PropertyDefinition class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyDefinition

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A property definition specifies the name, type, choice, and cardinality of values for properties.

propertyDefinitions

The values for this attribute are defined in Section 3.6.4.3.

3.6.4.3 Property Definitions

The PropertyDefinition class inherits property definitions from super classes.
The PropertyDefinition class MUST have the property definitions:

**icom_core:namespace**
- Description: Namespace for a property name.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:name**
- Description: Name for a property.
- Required: True
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_core:description**
- Description: A description of a property definition.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_meta:propertyType**
- Description: Type of a property.
- Required: False
- Inherited: False
- Property Type: icom_meta:PropertyType
- Cardinality: Single
- Updatability: On Create
- Choices: {PropertyChoiceType}
- Open Choice: False

Note: The notation \{PropertyChoiceType\} represents a set of PropertyChoiceType.

**icom_meta:defaultValue**
- Description: A default value for a property.
- Required: False
- Inherited: False
Property Type: **property-type**
Cardinality: Single
Updatability: Read Write

icom_meta:choice
Description: An allowed value for a property.
Required: False
Inherited: False
Property Type: icom_meta:PropertyChoiceType
Cardinality: Multi
Updatability: Read Write

icom_meta:openChoice
Description: Indicates whether value of the property must be listed among the choices.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_meta:inherited
Description: Indicates whether a property definition is inherited from a super class.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_meta:required
Description: Indicates whether a property value must be provided. It is applicable only when the updatability of the property is read-write or on-create.
Required: True
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_meta:updatability
Description: Updatability of a property specifying under what circumstances the property value can be updated.
Required: True
The PropertyDefinition class MAY include additional property definitions which are implementation-defined.

### 3.6.5 Property

#### 3.6.5.1 Description

The property holds a property value.

#### 3.6.5.2 Class Definition

The Property class has attribute values:

**localNamespace**

Value: icom_meta
localName
   Value: Property

extendsFrom
   Value:

stereotype
   Value: primary

description
   Value: A property value.

propertyDefinitions
   The values for this attribute are defined in Section 3.6.5.3.

3.6.5.3 Property Definitions
The Property class MUST have the property definitions:

icom_meta:propertyDefinition
   Description: A property definition that specifies the name, type, and
cardinality of a property.
   Required: True
   Inherited: False
   Property Type: icom_meta:PropertyDefinition
   Cardinality: Single
   Updatability: On Create

icom_meta:value
   Description: A value of a property.
   Required: True
   Inherited: False
   Property Type: property-type
   Cardinality: Single
   Updatability: Read Write

The Property class MAY include additional property definitions which are implementation-defined.
Figure 19: Property Definition and Property Class Diagram.

### 3.6.6 PropertyChoiceType

#### 3.6.6.1 Description

The property choice type represents a value choice for a property. Each choice includes a display name to be used for presentation purpose and a value to be stored in a property when a choice is selected.

#### 3.6.6.2 Class Definition

The PropertyChoiceType class has attribute values:

- `localNamespace`
  - Value: `icom_meta`
- `localName`
  - Value: `PropertyChoiceType`
- `extendsFrom`
  - Value:
stereotype
Value: primary
description
Value: A choice for a property value.
propertyDefinitions
The values for this attribute are defined Section 3.6.6.3.

3.6.6.3 Property Definitions
The PropertyChoiceType class MUST have the property definitions:

icom_core:description
Description: A description of a property choice.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_meta:displayName
Description: Display name of a property choice.
Required: True
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_meta:value
Description: A value of a property choice.
Required: True
Inherited: False
Property Type: property-type
Cardinality: Single
Updatability: Read Write
The PropertyChoiceType class MAY include additional property definitions which are implementation-defined.

3.6.7 PropertyType
A PropertyType expresses a name of a property-type.
3.6.7.1 Class Definition

The PropertyType class is a mixin class which expresses a name of a property-type.

The PropertyType class has attribute values:

localNamespace
  Value: icom_meta

localName
  Value: PropertyType

extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: PropertyType is a mixin class which expresses a name of a property-type.

propertyDefinitions
  The values for this attribute are defined in Section 3.6.7.2.

3.6.7.2 Property Definitions

The PropertyType class MAY include additional property definitions which are implementation-defined.

3.6.8 PropertyTypeEnum

The PropertyTypeEnum class is an enum class that enumerates the instances each of which expresses the name of a property-type.

The PropertyTypeEnum class has attribute values:

localNamespace
  Value: icom_meta

localName
  Value: PropertyTypeEnum

extendsFrom
  Value: icom_meta:PropertyType

stereotype
  Value: primary
isEnumeration
    Value: TRUE

description
    Value: Name of a basic data type.

instances

ICOM defines nine data types:

- icom_meta:String is equivalent to XML schema type xsd:string.
- icom_meta:Boolean is equivalent to XML schema type xsd:boolean.
- icom_meta:Decimal is equivalent to XML schema type xsd:decimal.
- icom_meta:Integer is equivalent to XML schema type xsd:integer.
- icom_meta:Datetime is equivalent to XML schema type xsd:dateTime.
- icom_meta:Duration is equivalent to XML schema type xsd:duration.
- icom_meta:IRI is equivalent to XML schema type xsd:anyURI.
- icom_meta:ID opaque object identifiers.
- icom_meta:HTML documents or fragments of Hypertext Markup Language (HTML) content


3.6.9 Updatability

3.6.9.1 Description
Updatability specifies under what circumstances a property value can be updated.

3.6.9.2 Class Definition
The Updatability class is a mixin class which specifies under what circumstances a property value can be updated.

The Updatability class has attribute values:

localNamespace
    Value: icom_meta

localName
    Value: Updatability

extendsFrom
    Value:
sterotype
  Value: mixin

description
  Value: Updatability is a mixin class which specifies under what circumstances a property value can be updated.

propertyDefinitions
  The values for this attribute are defined in Section 3.6.9.3.

3.6.9.3 Property Definitions
The Updatability class MAY include additional property definitions which are implementation-defined.

3.6.10 UpdatabilityEnum
The UpdatabilityEnum class is an enum class that enumerates instances each of which expresses the updatability of a property.
The UpdatabilityEnum has attribute values:

   localNamespace
     Value: icom_meta

   localName
     Value: UpdatabilityEnum

   extendsFrom
     Value: icom_meta:Updatability

   stereotype
     Value: primary

   isEnumeration
     Value: TRUE

   description
     Value: Updatability of a property.

   instances
     Value: <icom_meta:ReadOnly, icom_meta:WriteOnly, icom_meta:ReadWrite, icom_meta:OnCreate>

ICOM defines four updatability types:
- icom_meta:ReadOnly a property must not be updated directly by application.
- icom_meta:WriteOnly a property can be updated but cannot be read by application.
• icom_meta:ReadWrite a property can be updated by application.
• icom_meta:WriteOnly a property can be updated during the creation by application.

3.6.11 Cardinality

3.6.11.1 Description
Cardinality specifies whether a property is single or multi valued.

3.6.11.2 Class Definition
The Cardinality class is a mixin class which defines whether a property is single or multi valued.
The Cardinality class has attribute values:

localNamespace
    Value: icom_meta

localName
    Value: Cardinality

extendsFrom
    Value:

stereotype
    Value: mixin

description
    Value: Cardinality is a mixin class which defines whether a property is single or multi valued.

propertyDefinitions
    The values for this attribute are defined in Section 3.6.11.3.

3.6.11.3 Property Definitions
The Cardinality class MAY include additional property definitions which are implementation-defined.

3.6.12 CardinalityEnum
The CardinalityEnum class is an enum class that enumerates instances each of which expresses the
cardinality of a property.
The CardinalityEnum has attribute values:

localNamespace
    Value: icom_meta
ICOM defines two cardinality types:

- **icom_meta:Single** a property can have zero or one value (if property is not required), or exactly one value (if property is required).

- **icom_meta:Multi** a property can have zero or more values (if property is not required), or one or more values (if property is required).

### 3.6.13 Marker and Subclasses

*Figure 20: Marker Branch.*

Figure 20 depicts the main classes of Marker Branch, which includes Marker, Category, and Tag.
3.6.14 Marker

3.6.14.1 Description
A marker is an artifact that groups together entities by a criterion. Markers can be flat or hierarchical. Flat markers are modeled by tag and hierarchical markers are modeled by category.

Note: In some cases when a user applies a marker to an entity, the marker application should be private such that only the user who applies the marker can browse or locate the entity through the marker. This is especially the case when markers are created by a user and visible only to the user who created them.

3.6.14.2 Class Definition
The Marker class has attribute values:

    localNamespace
        Value: icom_meta

    localName
        Value: Marker

    extendsFrom
        Value: icom_core:Artifact

    stereotype
        Value: primary

    isAbstract
        Value: TRUE

    description
        Value: A marker is an artifact that groups together entities by a criterion.

    propertyDefinitions
        The values for this attribute are defined in Section 3.6.14.3.

3.6.14.3 Property Definitions
The Marker class inherits property definitions from super classes.

The Marker class MUST have the property definition:

    icom_meta:markedEntity
        Description: A marked entity.
        Required: False
        Inherited: False
        Property Type: icom_core:Entity
        Cardinality: Multi
        Updatability: Read Only
The Marker class MAY include additional property definitions which are implementation-defined.

Figure 21: Marker Class Diagram.

3.6.15 Category

3.6.15.1 Description
A category is a marker that classifies entities.

3.6.15.2 Class Definition
The Category class has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: Category

- **extendsFrom**
  - Value: icom_meta:Marker, icom_core:Container

- **stereotype**
  - Value: primary

- **description**
  - Value: A category is a marker that classifies entities.

3.6.15.3 Property Definitions
The Category class inherits property definitions from super classes. The Category class MUST have the property definitions:

- **icom_meta:superCategory**
  - Description: Zero or more super categories.
  - Required: False
Inherited: False
Property Type: icom_meta:Category
Cardinality: Multi
Updatability: Read Only

icom_meta:subCategory
Description: Zero or more sub categories.
Required: False
Inherited: False
Property Type: icom_meta:Category
Cardinality: Multi
Updatability: Read Only

icom_meta:abstract
Description: Indicates whether a category is abstract or concrete.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_meta:propertyDefinition
Description: Optional or mandatory properties for a category application.
Required: False
Inherited: False
Property Type: icom_meta:PropertyDefinition
Cardinality: Multi
Updatability: Read Write

The Category class MAY include additional property definitions which are implementation-defined.
3.6.16 CategoryApplication

3.6.16.1 Description

A category application is an instance of association between a category and a specific entity.

3.6.16.2 Class Definition

The CategoryApplication class has attribute values:

```
localNamespace
  Value: icom_meta

localName
  Value: CategoryApplication

extendsFrom
  Value: icom_core:Identifiable

stereotype
  Value: primary

description
  Value: A category application is an instance of association between a category and a specific entity.

propertyDefinitions
  The values for this attribute are defined in Section 3.6.16.3.
```
3.6.16.3 Property Definitions

The CategoryApplication class inherits property definitions from super classes. The CategoryApplication class MUST have the property definitions:

icom_meta:attachedEntity

- Description: An entity to which a category is applied.
- Required: True
- Inherited: False
- Property Type: icom_core:Entity
- Cardinality: Single
- Updatability: On Create

icom_meta:category

- Description: A category which is applied on an entity.
- Required: True
- Inherited: False
- Property Type: icom_meta:Category
- Cardinality: Single
- Updatability: On Create

icom_meta:property

- Description: Zero or more properties.
- Required: False
- Inherited: False
- Property Type: icom_meta:Property
- Cardinality: Multi
- Updatability: Read Write

The CategoryApplication class MAY include additional property definitions which are implementation-defined.

3.6.17 Tag

3.6.17.1 Description

A tag is a marker that labels entities by a keyword.

3.6.17.2 Class Definition

The Tag class has attribute values:

localNamespace

- Value: icom_meta

31 January 2013
Standards Track Work Product
Copyright © OASIS Open 2013. All Rights Reserved.
localName
Value: Tag

extendsFrom
Value: icom_meta:Marker

stereotype
Value: primary

description
Value: A tag is a marker that labels entities by a keyword.

propertyDefinitions
The values for this attribute are defined in Section 3.6.17.3.

3.6.17.3 Property Definitions
The Tag class inherits property definitions from super classes.
The Tag class MUST have the property definition:

icom_meta:applicationCount
Description: An estimate of the number of times a tag is applied on entities.
Required: False
Inherited: False
Property Type: Integer
Cardinality: Single
Updatability: Read Only

The Tag class MAY include additional property definitions which are implementation-defined.
3.6.18 TagApplication

3.6.18.1 Description

A tag application is an instance of association between a tag and a specific entity.

3.6.18.2 Class Definition

The TagApplication class has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: TagApplication

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: primary

- **description**
  - Value: A tag application is an instance of association between a tag and a specific entity.

3.6.18.3 Property Definitions

The TagApplication class inherits property definitions from super classes.

Figure 23: Tag and Tag Application Class Diagram.
The TagApplication class MUST have the property definitions:

### icom_meta:attachedEntity
- **Description:** An entity on which a tag is applied.
- **Required:** True
- **Inherited:** False
- **Property Type:** icom_core:Entity
- **Cardinality:** Single
- **Updatability:** On Create

### icom_meta:tag
- **Description:** A tag which is applied to an entity.
- **Required:** True
- **Inherited:** False
- **Property Type:** icom_meta:Tag
- **Cardinality:** Single
- **Updatability:** On Create

### icom_meta:appliedBy
- **Description:** A user who applies a tag to an entity.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:Actor
- **Cardinality:** Single
- **Updatability:** Read Only

### icom_meta:applicationDate
- **Description:** A date and time when a tag is applied to an entity.
- **Required:** False
- **Inherited:** False
- **Property Type:** DateTime
- **Cardinality:** Single
- **Updatability:** Read Write

The TagApplication class MAY include additional property definitions which are implementation-defined.

### 3.6.19 RelationshipBondable

#### 3.6.19.1 Description
A relationship bondable entity is an entity which may be related to other entities by a relationship.

Note: A relationship can exist among entities that are not relationships.
3.6.19.2 Class Definition

The RelationshipBondable class is a mixin class which defines the characteristics of entities that may be relationship bonded. It includes almost every subclass of Entity except Relationship.

The RelationshipBondable class has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: RelationshipBondable

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: mixin

- **description**
  - Value: RelationshipBondable is a mixin class which defines the characteristics of entities that can be relationship bonded.

3.6.19.3 Property Definitions

The RelationshipBondable class inherits property definitions from super classes. The RelationshipBondable class MAY include additional property definitions which are implementation-defined.

3.6.20 RelationshipDefinition

3.6.20.1 Description

A relationship definition is an entity that defines a type of relationship, including a name and a description of the relationship type, types of source entity and target entities of a relationship, and definition of properties in a relationship.

3.6.20.2 Class Definition

The RelationshipDefinition class has attribute values:

- **localNamespace**
  - Value: icom_meta
localName
  Value: RelationshipDefinition

extendsFrom
  Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype
  Value: primary

description
  Value: A relationship definition is an entity that defines a type of relationship.

propertyDefinitions
  The values for this attribute are defined in Section 3.6.20.3.

3.6.20.3 Property Definitions

The RelationshipDefinition class inherits property definitions from super classes.
The RelationshipDefinition class MUST have the property definitions:

icom_core:description
  Description: A description of a relationship definition.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write

icom_meta:propertyDefinition
  Description: Optional or mandatory properties for a relationship.
  Required: False
  Inherited: False
  Property Type: icom_meta:PropertyDefinition
  Cardinality: Multi
  Updatability: Read Write

icom_meta:allowedSourceType
  Description: A list of expanded names of relationship bondable classes, indicating that the source entity of a relationship MUST be an instance of a class in the list.
  Required: False
  Inherited: False
  Property Type: IRI
  Cardinality: Multi
3.6.21 Relationship

3.6.21.1 Description

A relationship is an entity that relates a set of entities by a predicate.

3.6.21.2 Class Definition

The Relationship class has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: Relationship

- **extendsFrom**
  - Value: icom_core:Entity

- **stereotype**
  - Value: primary

- **description**
  - Value: A relationship is an entity that relates a set of entities by a predicate.

3.6.21.3 Property Definitions

The Relationship class inherits property definitions from super classes.

- **icom_mta:allowedTargetType**
  - Description: A list of expanded names of relationship bondable classes, indicating that the target entity of a relationship MUST be an instance of a class in the list.
  - Required: False
  - Inherited: False
  - Property Type: IRI
  - Cardinality: Multi
  - Updatability: Read Write

The RelationshipDefinition class MAY include additional property definitions which are implementation-defined.
The Relationship class MUST have the property definitions:

**icom_meta:relationshipDefinition**
- Description: A definition of relationships.
- Required: True
- Inherited: False
- Property Type: icom_meta:RelationshipDefinition
- Cardinality: Single
- Updatability: On Create

**icom_meta:sourceEntity**
- Description: A source entity of a relationship.
- Required: True
- Inherited: False
- Property Type: icom_meta:RelationshipBondable
- Cardinality: Single
- Updatability: On Create

**icom_meta:targetEntity**
- Description: One or more target entities of a relationship.
- Required: True
- Inherited: False
- Property Type: icom_meta:RelationshipBondable
- Cardinality: Multi
- Updatability: Read Write

**icom_meta:property**
- Description: Zero or more properties.
- Required: False
- Inherited: False
- Property Type: icom_meta:Property
- Cardinality: Multi
- Updatability: Read Write

The Relationship class MAY include additional property definitions which are implementation-defined.
3.7 Common Concepts

3.7.1 Addressable

3.7.1.1 Description

An addressable object is an identifiable object that has one or more addresses.

3.7.1.2 Class Definition

The Addressable class is a mixin class which defines the characteristics of entities that has one or more addresses.

The Addressable class has attribute values:

- `localNamespace`
  - Value: `icom_core`

- `localName`
  - Value: `Addressable`
extendsFrom
    Value: icom_core:Identifiable

stereotype
    Value: mixin

description
    Value: Addressable is a mixin class which defines the characteristics of entities that has one or more addresses.

propertyDefinitions
    The values for this attribute are defined in Section 3.7.1.3.

3.7.1.3 Property Definitions
The Addressable class inherits property definitions from super classes.
The Addressable class MUST have the property definitions:

icom_core::entityAddress
    Description: Zero or more addresses of an addressable object.
    Required: False
    Inherited: False
    Property Type: icom_core::EntityAddress
    Cardinality: Multi
    Updatability: Read Write

icom_core::primaryAddress
    Description: The primary address of an addressable object.
    Required: False
    Inherited: False
    Property Type: icom_core::EntityAddress
    Cardinality: Single
    Updatability: Read Write

The Addressable class MAY include additional property definitions which are implementation-defined.

3.7.2 EntityAddress

3.7.2.1 Description
An entity address object represents an address which is defined by type and IRI.

3.7.2.2 Class Definition
The EntityAddress class has attribute values:
localNamespace
  Value: icom_core

localName
  Value: EntityAddress

extendsFrom
  Value:

stereotype
  Value: primary

description
  Value: An entity address object represents an address which is defined by type and IRI.

propertyDefinitions
  The values for this attribute are defined in Section 3.7.2.3.

3.7.2.3 Property Definitions

The EntityAddress class MUST have the property definitions:

icom_core:addressType
  Description: Type of an address.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write

icom_core:address
  Description: A IRI representing an address.
  Required: False
  Inherited: False
  Property Type: IRI
  Cardinality: Single
  Updatability: Read Write

3.7.3 Participant

3.7.3.1 Description

A participant object represents the participation of any addressable entity in a collaboration activity such as an occurrence, task, conference, discussion, and message.
If an addressable entity is not specified, an address must be specified.

### 3.7.3.2 Class Definition

The Participant class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Participant

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: A participant object represents the participation of any addressable entity in a collaboration activity such as an occurrence, task, conference, discussion, and message.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.7.3.3.

### 3.7.3.3 Property Definitions

The Participant class inherits property definitions from super classes.

The Participant class MUST have the property definitions:

- **icom_core:participant**
  - Description: An addressable entity to participate in a collaboration activity.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Addressable
  - Cardinality: Single
  - Updatability: On Create

- **icom_core:address**
  - Description: An address of a participant in a collaboration activity.
  - Required: False
  - Inherited: False
  - Property Type: IRI
  - Cardinality: Single
  - Updatability: On Create
3280  **icom_core:name**
3281  Description: Name of a participant in a collaboration activity.
3282  Required: False
3283  Inherited: False
3284  Property Type: String
3285  Cardinality: Single
3286  Updatability: On Create
3287
3288  The Participant class MAY include additional property definitions which are implementation-defined.
3289
3290  **3.7.4 Priority**
3291
3292  **3.7.4.1 Description**
3293  A priority level for delivery of information.
3294
3295  **3.7.4.2 Class Definition**
3296  The Priority class is a mixin class which defines a priority level for delivery of information.
3297  The Priority class has attribute values:
3298
3299  *localNamespace*  
3300  Value: icom_core
3301
3302  *localName*  
3303  Value: Priority
3304
3305  *extendsFrom*  
3306  Value:
3307
3308  *stereotype*  
3309  Value: mixin
3310
3311  *description*  
3312  Value: Priority is a mixin class which defines a priority level for delivery of information.
3313
3314  *propertyDefinitions*  
3315  The values for this attribute are defined in Section 3.7.4.3.
3316
3317  **3.7.4.3 Property Definitions**
3318  The Priority class MAY include additional property definitions which are implementation-defined.
3.7.5 PriorityEnum

The PriorityEnum class is an enum class that enumerates the instances each of which defines a priority level for delivery of information.

The PriorityEnum has attribute values:

- localNamespace
  Value: icom_core

- localName
  Value: PriorityEnum

- extendsFrom
  Value: icom_core:Priority

- stereotype
  Value: primary

- isEnumeration
  Value: TRUE

- description
  Value: Priority level for delivery of information.

- instances
  Value: <icom_core:Normal, icom_core:Low, icom_core:Medium, icom_core:High>

ICOM defines four priorities:

- icom_core:Normal a normal priority.
- icom_core:Low a low priority.
- icom_core:Medium a medium priority.
- icom_core:High a high priority.

3.7.6 DateTimeResolution

3.7.6.1 Description

A date time resolution is a resolution of date time value.

3.7.6.2 Class Definition

The DateTimeResolution class is a mixin class which defines a resolution of date time value.
The DateTimeResolution class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: DateTimeResolution

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

- **description**
  - Value: DateTimeResolution is a mixin class which defines a resolution of date time value.

**propertyDefinitions**
- The values for this attribute are defined in Section 3.7.6.3.

### 3.7.6.3 Property Definitions

The DateTimeResolution class MAY include additional property definitions which are implementation-defined.

### 3.7.7 DateTimeResolutionEnum

The DateTimeResolutionEnum class is an enum class that enumerates the instances each of which expresses a resolution of a date time value.

The DateTimeResolutionEnum has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: DateTimeResolutionEnum

- **extendsFrom**
  - Value: icom_core:DateTimeResolution

- **stereotype**
  - Value: primary
ICOM defines three date time resolutions:

- `icom_core:Year` date time resolution is in years.
- `icom_core:Date` date time resolution is in years and days.
- `icom_core:Time` date time resolution is in years, days, and time of day.

### 3.7.8 TimeZone

#### 3.7.8.1 Description

A time zone is a region that has a uniform standard time.

#### 3.7.8.2 Class Definition

The TimeZone class has attribute values:

- **localNamespace**
  - Value: `icom_core`

- **localName**
  - Value: `TimeZone`

- **extendsFrom**
  - Value: 

- **stereotype**
  - Value: `primary`

- **description**
  - Value: A time zone is a region that has a uniform standard time.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.7.8.3.

#### 3.7.8.3 Property Definitions

The TimeZone class inherits property definitions from super classes.
The TimeZone class MUST have the property definitions:

```

icom_core:ID
Description: Identifier of a time zone.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: On Create

icom_core:rawOffset
Description: An offset to add to Universal Coordinated Time (UTC) to get local time. If Daylight Saving Time is in effect at the specified date, the offset value is adjusted with the amount of daylight saving.
Required: False
Inherited: False
Property Type: Integer
Cardinality: Single
Updatability: On Create
```

The TimeZone class MAY include additional property definitions which are implementation-defined.

### 3.7.9 Location

#### 3.7.9.1 Description

A location object represents a physical location which is defined by name, description, and geo coordinates.

Note: The name of a location may remain unchanged while a physical location may be changing. For example, a location name might be “On an airplane” while a physical location might be the geo coordinates of a flight path or current coordinates of a plane.

#### 3.7.9.2 Class Definition

The Location class has attribute values:

```

class
localNamespace
Value: icom_core

localName
Value: Location

extendsFrom
Value:
```
3.7.9.3 Property Definitions

The Location class MUST have the property definitions:

- **icom_core:name**
  - Description: Name of a location.
  - Required: False
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:description**
  - Description: A description of a location.
  - Required: False
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:timeZone**
  - Description: Time zone of a location.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:TimeZone
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:coordinates**
  - Description: A list of geo coordinates marking a point, path, or area of a physical location.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:GeoCoordinates
The Location class MAY include additional property definitions which are implementation-defined.

3.7.10 GeoCoordinates

3.7.10.1 Description

A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

3.7.10.2 Class Definition

The GeoCoordinates class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: GeoCoordinates

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.7.10.3.

3.7.10.3 Property Definitions

The GeoCoordinates class MUST have the property definitions:

- **icom_core:latitude**
  - Description: Latitude of a location.
  - Required: False
  - Inherited: False
  - Property Type: Float
  - Cardinality: Single
  - Updatability: Read Write
icom_core:longitude

Description: Longitude of a location.
Required: False
Inherited: False
Property Type: Float
Cardinality: Single
Updatability: Read Write

icom_core:altitude

Description: Altitude of a location.
Required: False
Inherited: False
Property Type: Float
Cardinality: Single
Updatability: Read Write

The GeoCoordinates class MAY include additional property definitions which are implementation-defined.
4 Extension Modules

4.1 Overview of Extension Modules

Each extension module defines a model of a collaboration activity. Different models of collaboration activities in this specification include content creation, communication, coordination, discussion forum, and conference. Except for the Presence Module and Free Busy Module, the extension modules in this section introduce specialized subclasses of Artifact and Folder of Artifact Branch.

Note: ICOM Core Model (Section 3) establishes a framework to integrate specialized collaboration activities of the extension modules, which more or less represent technology or protocol channels. The framework is extensible with additional extension modules. For example, applications can adopt a model for CMIS Policy base type as a new extension module, which can be used to integrate with BPMN or BPEL processes outside the ICOM domain. An ICOM space can provide a durable context for continuity of conversations and activities related to a business process type or process instance. Some new extension modules may import the models from related standards. For example, social network model may be imported from [OpenGraph] or [OpenSocial].

Figure 25: Containers of Collaboration Activities.

ICOM defines containers that provide contexts and structures for specific areas of collaborative activities. The UML class diagram in Figure 25 depicts a Space as a hub of containers, including HeterogeneousFolder, AddressBook, Calendar, TaskList, Forum, and Conference. These containers are briefly described as follows:

HeterogeneousFolder (defined in Core Model) is a general purpose container that can contain any type of artifacts, and therefore, can serve as

- a library of documents and wiki pages to support content sharing and co-creation,
The following ten modules are specified as extension modules of ICOM:

1. **Content Module** (in Section 4.2) defines Content, MultiContent, and SimpleContent. A content represents a piece of data in a document or message. Content, multi-content, simple content, and online content form a composite design pattern.

2. **Document Module** (in Section 4.3) defines Document, WikiPage, and version control model. A document can contain a composite content defined in Section 4.2. Documents are typically contained by heterogeneous folders.

3. **Message Module** (in Section 4.4) defines Message, UnifiedMessage, InstantMessage, and related classes. A message can contain a composite content defined in Section 4.2. Unified messages are typically contained by heterogeneous folders.

4. **Presence Module** (in Section 4.5) defines Presence, Activity, and Contact Method. Presence represents a watchable state of a presentity (which is usually a person). Presence state is derived using an actor's subscriptions.

Note: Since a Presence is derived using a viewer's subscriptions, a Presence should not be shared with other viewers. For this reason, Presence is not modeled as Entity and is not assigned an access control list.

5. **Address Book Module** (in Section 4.6) defines AddressBook and PersonContact. A person contact can bookmark a reference to a person in an ICOM community as well as store addresses, phone numbers, and other entries about a person who may not be in any ICOM community.

6. **Calendar Module** (in Section 4.7) defines Calendar, Occurrence, and OccurrenceSeries. Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of meetings and booking of rooms and other resources.

7. **Free Busy Module** (in Section 4.8) defines FreeBusy. FreeBusy is a view derived from occurrences in a calendar or a set of calendars using an actor's privileges to determine the free or busy states of calendar occurrences.

Note: Since a FreeBusy view is derived using a viewer's privileges, a FreeBusy should not be shared with other viewers. For this reason, FreeBusy is not modeled as Entity and is not assigned an access control list.

8. **Task List Module** (in Section 4.9) defines TaskList and Task. Tasks are used to coordinate the assignment of tasks and to track the progress of task activities.

9. **Forum Module** (in Section 4.10) defines Forum, Topic, Announcement, and DiscussionMessage. Topics, announcements, and discussions are used for threaded discussions. Moderators of a forum can prune, merge, or fork the discussion threads.

10. **Conference Module** (in Section 4.11) defines Conference and related classes. A conference can contain visual, audio, and chat transcripts of the conference sessions. It also contains the current status, conference settings, past sessions, active session, and activity logs.
4.2 Content Module

4.2.1 MimeConvertible

4.2.1.1 Description
A MimeConvertible object represents an object that has Multipurpose Internet Mail Extensions (MIME) characteristics such as headers, content transfer encoding, and possible hierarchy of sub-contents.

4.2.1.2 Class Definition
The MimeConvertible class is a mixin class that defines the characteristics of objects that can be represented in MIME format.

The MimeConvertible class has attribute values:

- **localNamespace**
  - Value: icom_content

- **localName**
  - Value: MimeConvertible

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: mixin

- **description**
  - Value: MimeConvertible class is a mixin class that defines the characteristics of objects that can be represented in MIME format.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.2.1.3.

4.2.1.3 Property Definitions
The MimeConvertible class inherits property definitions from super classes.

The MimeConvertible class MAY include additional property definitions which are implementation-defined.

4.2.2 Content

4.2.2.1 Description
A content object represents a piece of data in a document or message. Content, multi-content, simple content, and online content form a composite design pattern.
4.2.2.2 Class Definition

The Content class has attribute values:

```plaintext
localNamespace
  Value: icom_content

localName
  Value: Content

extendsFrom
  Value: icom_core:Identifiable, icom_content:MimeConvertible

stereotype
  Value: primary

isAbstract
  Value: TRUE

description
  Value: Content represents a piece of data in a document or message.

propertyDefinitions
  The values for this attribute are defined in Section 4.2.2.3.
```

4.2.2.3 Property Definitions

The Content class inherits property definitions from super classes. The Content class MUST have the property definitions:

```plaintext
icom_content:contentId
  Description: A content id is a unique identifier for a part of content in multi-part contents.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write

icom_content:mediaType
  Description: Media type is a two-part identifier for Internet file formats.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
```
Updatability: Read Write

icom_content:contentDisposition

Description: Content disposition specifies a presentation style.
Required: False
Inherited: False
Property Type: icom_content:ContentDispositionType
Cardinality: Single
Updatability: Read Write

The Content class MAY include additional property definitions which are implementation-defined.

Figure 26: Composite Content Class Diagram.

4.2.3 MultiContent

4.2.3.1 Description

A multi-content object represents multiple parts of a message or document. It is a composite content that can contain a list of simple or composite contents.
### 4.2.3.2 Class Definition

The MultiContent class has attribute values:

- **localNamespace**
  - Value: icom_content

- **localName**
  - Value: MultiContent

- **extendsFrom**
  - Value: icom_content:Content

- **stereotype**
  - Value: primary

- **description**
  - Value: A multi-content object represents the multiple parts of a message or document.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.2.3.3.

### 4.2.3.3 Property Definitions

The MultiContent class inherits property definitions from super classes. The MultiContent class MUST have the property definitions:

- **icom_content:part**
  - Description: Zero or more parts of a hierarchical composite content.
  - Required: False
  - Inherited: False
  - Property Type: icom_content:MimeConvertible
  - Cardinality: Multi
  - Updatability: Read Write

The MultiContent class MAY include additional property definitions which are implementation-defined.

### 4.2.4 SimpleContent

#### 4.2.4.1 Description

A simple content holds a single piece of data.
4.2.4.2 Class Definition

The SimpleContent class has attribute values:

localNamespace
Value: icom_content

localName
Value: SimpleContent

extendsFrom
Value: icom_content:Content

stereotype
Value: primary

description
Value: A simple content holds a single piece of data.

propertyDefinitions
The values for this attribute are defined in Section 4.2.4.3.

4.2.4.3 Property Definitions

The SimpleContent class inherits property definitions from super classes.

The SimpleContent class MUST have the property definitions:

icom_content:characterEncoding
Description: Character encoding specifies character set of a content (a missing value means that a piece of content should be treated as binary or raw).
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_content:contentEncoding
Description: Content encoding specifies encoding of a piece of content.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write
4.2.5 OnlineContent

4.2.5.1 Description

An online content holds an online artifact attached to a document, message, or invitation.

Note: An online artifact must be rendered as an IRI when a message or invitation is delivered to external recipients.

4.2.5.2 Class Definition

The OnlineContent class has attribute values:

```plaintext
localNamespace
    Value: icom_content

localName
    Value: OnlineContent
```
extendsFrom
    Value:  icom_content:Content

stereotype
    Value:  primary

description
    Value:  An online content holds an online artifact attached to a message or invitation.

propertyDefinitions
    The values for this attribute are defined in Section 4.2.5.3.

4.2.5.3 Property Definitions

The OnlineContent class inherits property definitions from super classes.
The OnlineContent class MUST have the property definition:

icom_content:onlineAttachment
    Description:  An online artifact attached to a message.
    Required:  True
    Inherited:  False
    Property Type:  icom_core:Artifact
    Cardinality:  Single
    Updatability:  Read Write

The OnlineContent class MAY include additional property definitions which are implementation-defined.

4.2.6 ContentDispositionType

4.2.6.1 Description

A content disposition type is a presentation style of content.

4.2.6.2 Class Definition

The ContentDispositionType class is a mixin class which defines a presentation style of content.
The ContentDispositionType class has attribute values:

localNamespace
    Value:  icom_content

localName
    Value:  ContentDispositionType
4.2.6.3 Property Definitions

The ContentDispositionType class MAY include additional property definitions which are implementation-defined.

4.2.7 ContentDispositionTypeEnum

The ContentDispositionTypeEnum class is an enum class that enumerates the instances each of which expresses a presentation style of content.

The ContentDispositionTypeEnum class has attribute values:

- **localNamespace**
  - Value: icom_content

- **localName**
  - Value: ContentDispositionTypeEnum

- **extendsFrom**
  - Value: icom_content:ContentDispositionType

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A presentation style of content.

- **instances**
  - Value: <icom_content:Inline, icom_content:Attachment>
ICOM defines two content disposition types:

- **icom_content:Inline** content is to be displayed automatically upon display of the main body of an artifact.
- **icom_content:Attachment** content is separate from the main body of an artifact, and that its display should not be automatic, but contingent upon some further action of a user.

### 4.2.8 AttachedItem

#### 4.2.8.1 Description

An attached item holds a content for an occurrence, task, and contact artifact.

#### 4.2.8.2 Class Definition

The AttachedItem class has attribute values:

- **localNamespace**
  - Value: icom_content

- **localName**
  - Value: AttachedItem

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: An attachedItem holds a content for an occurrence, task, and contact artifact.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.2.8.3.

#### 4.2.8.3 Property Definitions

The AttachedItem class MUST have the property definitions:

- **icom_core:name**
  - Description: Name of a content attachment.
  - Required: True
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write
4.3 Document Module

4.3.1 Versionable

4.3.1.1 Description

A versionable artifact is

1. a non-version-controlled copy,
2. a specific versioned copy,
3. a private working copy, or
4. a representative copy (optional)

of an artifact version series.

When a versionable artifact is not under version control, a non-version-controlled copy MUST be the only copy in a version series, i.e. there is only one copy and one objectId.

When a versionable artifact is under version control, a representative copy MAY provide a version-independent view of a versionable artifact.

When a non-version-controlled copy is placed under version control, a versioned copy MUST be created. Assignment of an object identifier to a versioned copy is implementation-dependent:

- if a versioned copy retains the object identifier of a non-version-controlled copy, the version type of a versionable artifact MUST change from NonVersionControlledCopy to VersionedCopy;
- if a versioned copy is assigned a new object identifier that is different from the object identifier of a non-version-controlled copy, a representative copy MAY retain the object identifier of the non-version-controlled copy;
- if both versioned copy and representative copy are assigned new object identifiers that are different from the object identifier of a non-version-controlled copy, the non-version-controlled copy SHALL be discarded.

When a private working copy is checked in, a versioned copy MUST be created. Assignment of an object identifier to a versioned copy is implementation-dependent:

- if a versioned copy retains the object identifier of a private working copy, the version type of a versionable artifact MUST change from PrivateWorkingCopy to VersionedCopy;
- if a versioned copy is assigned a new object identifier that is different from the object identifier of a private working copy, the private working copy SHALL be discarded.

It is optional for a service provider to provide a representative copy for a version series. If a representative copy is provided:

- a representative copy MUST have its own object identifier that is different from the object identifier of any versioned copy or private working copy;
- assignment of an object identifier to a representative copy is implementation-dependent:
  - a representative copy MAY retain the object identifier of a non-version-controlled copy; if so the version type of a versionable artifact MUST change from NonVersionControlledCopy to RepresentativeCopy;
  - a representative copy MAY be assigned a new object identifier that is different from the object identifier of a non-version-controlled copy;
- content and state of a representative copy is implementation-dependent:
  - a representative copy MAY be a copy of the content and state of the latest versioned copy or the latest major versioned copy in a version series;
  - a representative copy MAY be a copy of the content and state of a private working copy if the current user loading the representative copy is the same user who checks out a version series.

Note: Each versioned copy of a versionable artifact is itself a versionable artifact, i.e. it has its own objectID. A versioned copy has a version number, label, and check in comment.

Note: A private working copy is a versionable artifact created by an explicit checkout operation on a versionable artifact under version control. The properties for a private working copy are identical to the properties of a versioned copy on which a checkout operation was performed. Certain properties such as objectID and creationDate are different from a versioned copy. The content of a private working copy is identical to the content of a versioned copy. Its object identifier is different from that of the representative copy or any versioned copy.

A private working copy MAY be saved in a version series for sharing and co-editing, however, it needs not be visible to users who may only have permissions to view other versioned copies in a version series.

Note: Until it is checked in using an explicit check-in operation, a private working copy must not be considered the LatestMajorVersion in a version series.

A container of a versionable artifact CAN contain a representative copy so that it provides a version-independent view of a state of the version series.

Note: Starting from a representative copy in a container, an actor can traverse a version series to retrieve any versioned copy or private working copy.

ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of Content Management Interoperability Services Version 1.0 [CMIS].

### 4.3.1.2 Class Definition

The Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.

The Versionable class has attribute values:

```plaintext
localNamespace
  Value: icom_doc

localName
  Value: Versionable

extendsFrom
  Value: icom_core:Identifiable

stereotype
  Value: mixin
```
Value: Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.

propertyDefinitions

The values for this attribute are defined in Section 4.3.1.3.

4.3.1.3 Property Definitions

The Versionable class inherits property definitions from super classes.
The Versionable class MUST have the property definitions:

icom_doc:versionControlMetadata

Description: A version control metadata object attached to a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:VersionControlMetadata
Cardinality: Single
Updatability: Read Only

icom_doc:versionType

Description: A type of version controlled copy of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:VersionType
Cardinality: Single
Updatability: Read Only

The Versionable class MAY include additional property definitions which are implementation-defined.

4.3.2 VersionControlMetadata

4.3.2.1 Description

A version control metadata is an object that contains version control information.
There are two classes of version control metadata: version series and version. A version control metadata of a versionable artifact is either a version series or a version depending on the version type.

- If the version type is icom_doc:NonVersionControlledCopy then metadata is optional; if metadata is present, it MUST be a version series object.
- If the version type is icom_doc:RepresentativeCopy, then metadata MUST be a version series object.
- If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata MUST be a version object.
4.3.2.2 Class Definition

The VersionControlMetadata class is a mixin class that defines the characteristics of version or version series metadata for version control.

The VersionControlMetadata class has attribute values:

- `localNamespace`
  - Value: `icom_doc`

- `localName`
  - Value: `VersionControlMetadata`

- `extendsFrom`
  - Value: `icom_core:Identifiable`

- `stereotype`
  - Value: `mixin`

- `description`
  - Value: `VersionControlMetadata is a mixin class that defines the characteristics of entities that serve as metadata for version control.`

- `propertyDefinitions`
  - The values for this attribute are defined in Section 4.3.2.3.

4.3.2.3 Property Definitions

The VersionControlMetadata class inherits property definitions from super classes.

The VersionControlMetadata class MUST have the property definition:

- `icom_doc:representativeCopy`
  - Description: A representative copy of a versionable artifact.
  - Required: False
  - Inherited: False
  - Property Type: `icom_doc:Versionable`
  - Cardinality: Single
  - Updatability: Read Only

The VersionControlMetadata class MAY include additional property definitions which are implementation-defined.
4.3.3 VersionSeries

4.3.3.1 Description
A version series is a version control metadata that contains a version history and check in/out states of a versionable artifact.

A version series object is a version control metadata of a representative copy of a versionable artifact.

4.3.3.2 Class Definition
The VersionSeries class has attribute values:

   localNamespace
     Value: icom_doc

   localName
     Value: VersionSeries

   extendsFrom
     Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

   stereotype
     Value: primary

   description
     Value: A version series is version control metadata that contains a version history and check in/out states of a versionable artifact.

   propertyDefinitions
     The values for this attribute are defined in Section 4.3.3.3.

4.3.3.3 Property Definitions
The VersionSeries class inherits property definitions from super classes.

The VersionSeries class MUST have the property definitions:

   icom_doc:versionHistory
     Description: A history of version nodes of a versionable artifact.
     Required: False
     Inherited: False
     Property Type: icom_doc:Version
     Cardinality: Multi
     Updatability: Read Only

   icom_doc:versionableHistory
     Description: A history of the versioned copies of a versionable artifact.
<table>
<thead>
<tr>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>icom_doc:Versionable</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_doc:Versionable</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_core:Actor</td>
<td>Single</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_doc:latestVersionedCopy
- **Description**: Latest versioned copy of a versionable artifact.
- **Required**: False
- **Inherited**: False
- **Property Type**: icom_doc:Versionable
- **Cardinality**: Single
- **Updatability**: Read Only

### icom_doc:privateWorkingCopy
- **Description**: A private working copy of a versionable artifact.
- **Required**: False
- **Inherited**: False
- **Property Type**: icom_doc:Versionable
- **Cardinality**: Single
- **Updatability**: Read Only

### icom_doc:versionSeriesCheckedOut
- **Description**: Indicates whether a version series is checked out.
- **Required**: False
- **Inherited**: False
- **Property Type**: Boolean
- **Cardinality**: Single
- **Updatability**: Read Only

### icom_doc:versionSeriesCheckedOutBy
- **Description**: An actor who checks out a version series.
- **Required**: False
- **Inherited**: False
- **Property Type**: icom_core:Actor
- **Cardinality**: Single
- **Updatability**: Read Only

### icom_doc:versionSeriesCheckedOutOn
- **Description**: The time when a version series is checked out.
- **Required**: False
- **Inherited**: False
The VersionSeries class MAY include additional property definitions which are implementation-defined.

### 4.3.4 Version

#### 4.3.4.1 Description

A version is a version control metadata that contains a version number, label, and description. A version object is a version control metadata of a versioned copy or a private working copy of a versionable artifact.

#### 4.3.4.2 Class Definition

The Version class has attribute values:

- **localNamespace**
  - Value: icom_doc

- **localName**
  - Value: Version

- **extendsFrom**
  - Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable
**stereotype**
- Value: primary

**description**
- Value: A version is a version control metadata that contains a version number, label, and description.

**propertyDefinitions**
- The values for this attribute are defined in Section 4.3.4.3.

### 4.3.4.3 Property Definitions

The Version class inherits property definitions from super classes.
The Version class MUST have the property definitions;

**icom_doc:checkinComment**
- Description: A check in comment of a versioned copy.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_doc:versionNumber**
- Description: A version number of a versioned copy.
- Required: True
- Inherited: False
- Property Type: Integer
- Cardinality: Single
- Updatability: Read Write

**icom_doc:versionLabel**
- Description: A version label of a versioned copy.
- Required: True
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

**icom_doc:majorVersion**
- Description: Indicates whether a versioned copy is a major version.
- Required: True
- Inherited: False
- Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_doc:versionedOrPrivateWorkingCopy
Description: A versioned copy or private working copy corresponding to a
version of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:Versionable
Cardinality: Single
Updatability: Read Only

The Version class MAY include additional property definitions which are implementation-defined.

4.3.5 VersionType

4.3.5.1 Description
A version type is a version state of a copy of versionable document.

4.3.5.2 Class Definition
The VersionType class is a mixin class which defines a version state of a copy of versionable document.
The VersionType class has attribute values:

localNamespace
Value: icom_doc

localName
Value: VersionType

extendsFrom
Value: 

stereotype
Value: mixin

description
Value: VersionType is a mixin class which defines a version state of a copy of versionable
document.

propertyDefinitions
The values for this attribute are defined in Section 4.3.5.3.
4.3.5.3 Property Definitions

The VersionType class MAY include additional property definitions which are implementation-defined.

4.3.6 VersionTypeEnum

The VersionTypeEnum class is an enum class that enumerates the instances each of which expresses a version type.

The VersionTypeEnum class has attribute values:

- `localNamespace`
  
  Value: `icom_doc`

- `localName`
  
  Value: `VersionTypeEnum`

- `extendsFrom`
  
  Value: `icom_doc:VersionType`

- `stereotype`
  
  Value: `primary`

- `isEnumeration`
  
  Value: `TRUE`

- `description`
  
  Value: A version type of a copy of versionable document.

- `instances`
  
  Value: `<icom_doc:NonVersionControlledCopy, icom_doc:VersionedCopy, icom_doc:PrivateWorkingCopy, icom_doc:RepresentativeCopy>`

ICOM defines four version types:

- `icom_doc:NonVersionControlledCopy` a versionable artifact is not under version control.
- `icom_doc:VersionedCopy` a versionable artifact is a version of an artifact version series.
- `icom_doc:PrivateWorkingCopy` a versionable artifact is a private working copy of an artifact version series.
- `icom_doc:RepresentativeCopy` a versionable artifact is a version-independent representative copy of an artifact. This version type is optional and implementation-dependent.
4.3.7 Document

4.3.7.1 Description
A document is a versionable artifact that can contain a single content of a media type or composite contents of an assortment of media types.

4.3.7.2 Class Definition
The Document class has attribute values:

- `localNamespace`
  Value: icom_doc

- `localName`
  Value: Document

- `extendsFrom`
  Value: icom_core:Artifact, icom_doc:Versionable, icom_content:MimeConvertible

- `stereotype`
  Value: primary

- `description`
  Value: A document is a versionable artifact that may contain a single content of a media type or composite contents of an assortment of media types.

- `propertyDefinitions`
  The values for this attribute are defined in Section 4.3.7.3.

4.3.7.3 Property Definitions
The Document class inherits property definitions from super classes.

- `icom_content:content`
  Description: Content of a document.
  Required: False
  Inherited: False
  Property Type: icom_content:Content
  Cardinality: Single
  Updatability: Read Write

- `icom_doc:size`
  Description: The size of a copy of a document.
  Required: False
The Document class MAY include additional property definitions which are implementation-defined.

**Figure 27**: Document, Version Series, and Version Class Diagram.

### 4.3.8 WikiPage

#### 4.3.8.1 Description

A wiki page is a document that contains a wiki content and that provides an html page generated from the wiki content.
4.3.8.2 Class Definition

The WikiPage class has attribute values:

   localNamespace
      Value: icom_doc

   localName
      Value: WikiPage

   extendsFrom
      Value: icom_doc:Document

   stereotype
      Value: primary

   description
      Value: A wiki page is a document that contains a wiki content and that provides an html page
generated from the wiki content.

propertyDefinitions
   The values for this attribute are defined in Section 4.3.8.3.

4.3.8.3 Property Definitions

The WikiPage class inherits property definitions from super classes.
The WikiPage class MUST have the property definitions:

   icom_doc:renderedPage
      Description: An html page generated from a wiki content.
      Required: False
      Inherited: False
      Property Type: String
      Cardinality: Single
      Updatability: Read Only

The WikiPage class MAY include additional property definitions which are implementation-defined.
4.4 Message Module

4.4.1 Message

4.4.1.1 Description

A message is a unit of conversation. It holds a simple content or multipart message contents in a content property. It has a single sender.

Note: The delivered time is the time when a message is delivered to a given recipient. The user creation date and time property can be used as the sent date and time of a message. The name property can be used as the subject of a message.

4.4.1.2 Class Definition

The Message class has attribute values:

```
localNamespace
  Value: icom_msg

localName
  Value: Message

extendsFrom
  Value: icom_core:Artifact

stereotype
  Value: primary

isAbstract
  Value: TRUE
```
description
Value: A message is a unit of conversation.

propertyDefinitions
The values for this attribute are defined in Section 4.4.1.3.

4.4.1.3 Property Definitions
The Message class inherits property definitions from super classes. The Message class MUST have the property definitions:

icom_content:content
Description: Content of a message
Required: False
Inherited: False
Property Type: icom_content:Content
Cardinality: Single
Updatability: Read Write

icom_msg:sender
Description: Sender of a message.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Single
Updatability: Read Write

icom_msg:deliveredTime
Description: The date and time when a message is delivered to a given recipient.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Only

The Message class MAY include additional property definitions which are implementation-defined.

4.4.2 UnifiedMessage

4.4.2.1 Description
A unified message is a type of message delivered electronically over a computer, voice, fax, and other networks.
A unified message can be one of these types:

- Email is a message delivered electronically over a computer network.
- Voice is a message that contains a voice or audio stream.
- Fax is a message that contains an image transmitted via phone lines using the fax protocol.
- Notification is a type of message sent by applications.

### 4.4.2.2 Class Definition

The UnifiedMessage class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessage

- **extendsFrom**
  - Value: icom_msg:Message, icom_content:MimeConvertible

- **stereotype**
  - Value: primary

- **description**
  - Value: A unified message is a type of message delivered electronically over a computer, voice, fax, and other networks.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.4.2.3.

### 4.4.2.3 Property Definitions

The UnifiedMessage class inherits property definitions from super classes.

The UnifiedMessage class MUST have the property definitions:

- **icom_core:prioritiy**
  - Description: The priority of a message.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Priority
  - Cardinality: Single
  - Updatability: Read Write

- **icom_content:contentId**
  - Description: Content id is a unique identifier for a message part in multi-part messages.
icom_content:mediaType
- Description: Media type is a two-part identifier for Internet file formats.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Read Write

icom_content:contentDisposition
- Description: Content disposition specifies a presentation style.
- Required: False
- Inherited: False
- Property Type: icom_content:ContentDispositionType
- Cardinality: Single
- Updatability: Read Write

icom_msg:envelopeSender
- Description: An envelope sender is a participant to receive bounced message. It is also known as return path.
- Required: False
- Inherited: False
- Property Type: icom_core:Participant
- Cardinality: Single
- Updatability: Read Write

icom_msg:toReceivers
- Description: A list of participants to receive a message.
- Required: False
- Inherited: False
- Property Type: icom_core:Participant
- Cardinality: Multi
- Updatability: Read Write

icom_msg:ccReceivers
- Description: A list of participants to receive carbon-copies of a message.
- Required: False
- Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
Updatability: Read Write

icom_msg:bccReceivers
Description: A list of participants to receive blind-carbon-copies of a message.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
Updatability: Read Write

icom_msg:replyTo
Description: A list of participants to receive a reply message.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
Updatability: Read Write

icom_msg:flag
Description: Zero or more flags on a message.
Required: False
Inherited: False
Property Type: icom_msg:UnifiedMessageFlag
Cardinality: Multi
Updatability: Read Write

icom_msg:messageDispositionNotificationRequested
Description: A message disposition notification requested for a message.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_msg:messageDeliveryStatusNotificationRequest
Description: Indicates the types of delivery status notifications requested for a message. Default is icom_msg:Failure.
Required: False
Inherited: False
Property Type: icom_msg:UnifiedMessageDeliveryStatusNotificationRequest
Cardinality: Multi
Updatability: Read Write

icom_msg:channel
Description: Indicates the delivery channel of a message.
Required: False
Inherited: False
Property Type: icom_msg:UnifiedMessageChannel
Cardinality: Single
Updatability: Read Write

icom_msg:editMode
Description: Indicates an editable mode (new, draft, or delivered) of a message.
Required: False
Inherited: False
Property Type: icom_msg:UnifiedMessageEditMode
Cardinality: Single
Updatability: Read Only

icom_msg:mimeHeader
Description: A list of headers. Each header is represented by a multi-valued property.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

icom_msg:size
Description: The size of a unified message.
Required: False
Inherited: False
Property Type: Integer
Cardinality: Single
Updatability: Read Only

The UnifiedMessage class MAY include additional property definitions which are implementation-defined.
4.4.3 UnifiedMessageParticipant

4.4.3.1 Description
A unified message participant object represents the participation of an addressable entity in a unified message.

4.4.3.2 Class Definition
The UnifiedMessageParticipant class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageParticipant

- **extendsFrom**
  - Value: icom_core:Participant

- **stereotype**
  - Value: primary

- **description**
  - Value: A unified message participant object represents the participation of an addressable entity in a unified message.

4.4.3.3 Property Definitions
The UnifiedMessageParticipant class inherits property definitions from super classes.
The UnifiedMessageParticipant class MUST have the property definitions:

- **icom_msg:fullAddress**
  - Description: Full address of a participant.
  - Required: False
  - Inherited: False
  - Property Type: IRI
  - Cardinality: Single
  - Updatability: Read Write

- **icom_msg:localPart**
  - Description: Local part of a full address.
  - Required: False
The UnifiedMessageParticipant class MAY include additional property definitions which are implementation-defined.

### 4.4.4 UnifiedMessageFlag

#### 4.4.4.1 Description

A unified message flag is a flag on a message.

#### 4.4.4.2 Class Definition

The UnifiedMessageFlag class is a mixin class which defines a flag on a message. The UnifiedMessageFlag class has attribute values:

```plaintext
localNamespace
  Value: icom_msg

localName
  Value: UnifiedMessageFlag

extendsFrom
  Value: 

stereotype
  Value: mixin

description
  Value: UnifiedMessageFlag is a mixin class which defines a flag on a message.

propertyDefinitions
  The values for this attribute are defined in Section 4.4.4.3.
```
4.4.3 Property Definitions

The UnifiedMessageFlag class MAY include additional property definitions which are implementation-defined.

4.4.5 UnifiedMessageFlagEnum

The UnifiedMessageFlagEnum class is an enum class that enumerates the instances each of which expresses a flag on a message.

The UnifiedMessageFlagEnum class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageFlagEnum

- **extendsFrom**
  - Value: icom_msg:UnifiedMessageFlag

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A flag on a message.

- **instances**

ICOM defines eight flags:

- **icom_msg:Answered** a message is answered.
- **icom_msg:Forwarded** a message is forwarded.
- **icom_msg:Redirected** a message is redirected.
- **icom_msg:Hidden** a message is hidden.
- **icom_msg:MarkedForDelete** a message is marked for delete.
- **icom_msg:MarkedForFollowUp** a message is marked for follow up.
- **icom_msg:MarkedForDraft** a message is marked for draft.
- **icom_msg:MessageDispositionNotificationProcessed** a message disposition notification is processed.
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest

4.4.6.1 Description

A unified message delivery status notification request is a directive for notifying a participant of delivery status of a message.

4.4.6.2 Class Definition

The UnifiedMessageDeliveryStatusNotificationRequest class is a mixin class which defines a directive for notifying a participant of delivery status of a message.

The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:

- `localNamespace`
  - Value: `icom_msg`

- `localName`
  - Value: `UnifiedMessageDeliveryStatusNotificationRequest`

- `extendsFrom`
  - Value:

- `stereotype`
  - Value: `mixin`

- `description`
  - Value: `UnifiedMessageDeliveryStatusNotificationRequest is a mixin class which defines a directive for notifying a participant of delivery status of a message.`

- `propertyDefinitions`
  - The values for this attribute are defined in Section 4.4.6.3.

4.4.6.3 Property Definitions

The UnifiedMessageDeliveryStatusNotificationRequest class MAY include additional property definitions which are implementation-defined.

4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum

The UnifiedMessageDeliveryStatusNotificationRequestEnum class is an enum class that enumerates the instances each of which expresses a request for one of several types of delivery status notification.

The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

- `localNamespace`
  - Value: `icom_msg`
ICOM defines four delivery status notification requests:

- **icom_msg:Never** a sender requests status notification not be returned to the sender under any condition.
- **icom_msg:Success** a sender requests a status notification for successful delivery of a message.
- **icom_msg:Failure** a sender requests a status notification for delivery failure of a message.
- **icom_msg:Delay** a sender requests a status notification when delivery of a message has been delayed for an unusual length of time.

### 4.4.8 UnifiedMessageChannel

#### 4.4.8.1 Description

A message channel used to deliver a unified message.

#### 4.4.8.2 Class Definition

The UnifiedMessageChannel class is a mixin class which defines a channel used to deliver a unified message.

The UnifiedMessageChannel class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageChannel
extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: UnifiedMessageChannel is a mixin class which defines a channel used to deliver a
  unified message.

propertyDefinitions
  The values for this attribute are defined in Section 4.4.8.3.

4.4.8.3 Property Definitions
  The UnifiedMessageChannel class MAY include additional property definitions which are implementation-
  defined.

4.4.9 UnifiedMessageChannelEnum
  The UnifiedMessageChannelEnum class is an enum class that enumerates the instances each of which
  expresses a type of delivery channel.
  The UnifiedMessageChannelEnum class has attribute values:

  localNamespace
    Value: icom_msg

  localName
    Value: UnifiedMessageChannelEnum

  extendsFrom
    Value: icom_msg:UnifiedMessageChannel

  stereotype
    Value: primary

  isEnumeration
    Value: TRUE

  description
    Value: A delivery channel.

  instances
    Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>
ICOM defines four channel types:

- **icom_msg:** Email delivery channel is email.
- **icom_msg:** Voice delivery channel is voice.
- **icom_msg:** Fax delivery channel is fax.
- **icom_msg:** Notification delivery channel is notification.

### 4.4.10 UnifiedMessageEditMode

#### 4.4.10.1 Description

A unified message edit mode is a mode that indicates whether a unified message is editable.

#### 4.4.10.2 Class Definition

The UnifiedMessageEditMode class is a mixin class which defines a mode that indicates whether a unified message is editable.

The UnifiedMessageEditMode class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageEditMode

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

- **description**
  - Value: UnifiedMessageEditMode is a mixin class which defines a mode that indicates whether a unified message is editable.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.4.10.3.

#### 4.4.10.3 Property Definitions

The UnifiedMessageEditMode class MAY include additional property definitions which are implementation-defined.

### 4.4.11 UnifiedMessageEditModeEnum

The UnifiedMessageEditModeEnum class is an enum class that enumerates the instances each of which expresses whether a message is a new copy, saved draft copy, or delivered copy.
The UnifiedMessageEditModeEnum class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageEditModeEnum

- **extendsFrom**
  - Value: icom_msg:UnifiedMessageEditMode

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A message is a new copy, a saved draft copy, or a delivered copy. New or draft copies are usually editable while delivered copies are usually not editable.

- **instances**
  - Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy>

ICOM defines three modes:

- **icom_msg:NewCopy** a message is a new message.
- **icom_msg:DraftCopy** a message is saved as a draft.
- **icom_msg:DeliveredCopy** a message is a sent or received message.
4.4.12 InstantMessage

4.4.12.1 Description

An instant message is a type of message for synchronous, usually text based, conversation.

4.4.12.2 Class Definition

The InstantMessage class has attribute values:

```
localNamespace
  Value: icom_msg

localName
  Value: InstantMessage

extendsFrom
  Value: icom_msg:Message
```
stereotype
   Value: primary

isAbstract
   Value: FALSE

description
   Value: An instant message is a type of message for synchronous, usually text based, conversation.

propertyDefinitions
   The values for this attribute are defined in Section 4.4.12.3.

4.4.12.3 Property Definitions
The InstantMessage class inherits property definitions from super classes.
The InstantMessage class MUST have the property definitions:

icom_msg:toReceivers
   Description: A list of participants to receive a message.
   Required: False
   Inherited: False
   Property Type: icom_core:Participant
   Cardinality: Multi
   Updatability: Read Write

icom_msg:conversationId
   Description: An identifier of a conversation involving one or more instant messages.
   Required: False
   Inherited: False
   Property Type: Integer
   Cardinality: Single
   Updatability: Read Write

icom_msg:clientSideld
   Description: An identifier of a client.
   Required: False
   Inherited: False
   Property Type: String
   Cardinality: Single
   Updatability: Read Write
icom_msg:formattingStyle
5085  Description: A style for formatting a rich text message.
5086  Required: False
5087  Inherited: False
5088  Property Type: String
5089  Cardinality: Single
5090  Updatability: Read Write
5091
5092
icom_msg:instantMessageType
5093  Description: A type of instant message.
5094  Required: False
5095  Inherited: False
5096  Property Type: icom_msg:InstantMessageType
5097  Cardinality: Single
5098  Updatability: Read Write
5099
5100
icom_msg:chatStatus
5101  Description: A chat status of a user.
5102  Required: False
5103  Inherited: False
5104  Property Type: icom_msg:InstantMessageChatStatus
5105  Cardinality: Single
5106  Updatability: Read Write
5107
5108
4.4.13 InstantMessageType

4.4.13.1 Description
An instant message type.

4.4.13.2 Class Definition
The InstantMessageType class is a mixin class which defines a type of instant message. The InstantMessageType class has attribute values:

```
   localNamespace
       Value: icom_msg

   localName
       Value: InstantMessageType

   extendsFrom
       Value: 
```
stereotype
    Value: mixin

description
    Value: InstantMessageType is a mixin class which defines a type of instant message.

propertyDefinitions
    The values for this attribute are defined in Section 4.4.13.3.

4.4.13.3 Property Definitions
The InstantMessageType class MAY include additional property definitions which are implementation-defined.

4.4.14 InstantMessageTypeEnum
The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which expresses a type of instant message.

The InstantMessageTypeEnum class has attribute values:

    localNamespace
        Value: icom_msg

    localName
        Value: InstantMessageTypeEnum

    extendsFrom
        Value: icom_msg:InstantMessageType

    stereotype
        Value: primary

    isEnumeration
        Value: TRUE

    description
        Value: A type of instant message.

    instances
        Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:Gone, icom_msg:OtherInstantMessageType>

ICOM defines five instant message types:

    • icom_msg:System an instant message is a system message.
• **icom_msg:Chat** an instant message is a chat message.
• **icom_msg:Broadcast** an instant message is a broadcast message.
• **icom_msg:Gone** an instant message is a message indicating that a user is gone.
• **icom_msg:OtherInstantMessageType** an instant message is of other type.

### 4.4.15 InstantMessageChatStatus

#### 4.4.15.1 Description

An instant message chat status defines a vocabulary of chat status.

#### 4.4.15.2 Class Definition

The `InstantMessageChatStatus` class is a mixin class which defines a chat status.

The `InstantMessageChatStatus` class has attribute values:

- **localNamespace**
  - Value: `icom_msg`

- **localName**
  - Value: `InstantMessageChatStatus`

- **extendsFrom**
  - Value: 

- **stereotype**
  - Value: `mixin`

- **description**
  - Value: `InstantMessageChatStatus is a mixin class which defines a chat status.`

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.4.15.3.

#### 4.4.15.3 Property Definitions

The `InstantMessageChatStatus` class MAY include additional property definitions which are implementation-defined.

### 4.4.16 InstantMessageChatStatusEnum

The `InstantMessageChatStatusEnum` class is an enum class that enumerates the instances each of which expresses a chat status of a user.

The `InstantMessageChatStatusEnum` class has attribute values:
ICOM defines five chat status:

- **icom_msg:Active** a user is active.
- **icom_msg:Typing** a user is typing.
- **icom_msg:Paused** a user has paused.
- **icom_msg:Inactive** a user is inactive.
- **icom_msg:Gone** a user is gone.

### 4.4.17 InstantMessageFeed

#### 4.4.17.1 Description

An instant message feed contains a set of instant message connections and a queue of outbound instant messages.

#### 4.4.17.2 Class Definition

The InstantMessageFeed class has attribute values:

- **localNamespace**
  - Value: icom_msg
- **localName**
  - Value: InstantMessageFeed
extendsFrom
   Value: icom_core:Entity

 stereotype
    Value: primary

description
    Value: An instant message feed contains a set of instant message connections and a queue of outbound instant messages.

propertyDefinitions
    The values for this attribute are defined in Section 4.4.17.3.

4.4.17.3 Property Definitions

The InstantMessageFeed class inherits property definitions from super classes.

The InstantMessageFeed class MUST have the property definitions:

icom_msg:connection
   Description: One or more instant messaging connections.
   Required: False
   Inherited: False
   Property Type: icom_msg:InstantMessageConnection
   Cardinality: Multi
   Updatability: Read Only

icom_msg:outboundInstantMessage
   Description: Outbound instant messages.
   Required: False
   Inherited: False
   Property Type: icom_msg:InstantMessage
   Cardinality: Multi
   Updatability: Write Only
4.4.18 InstantMessageConnection

4.4.18.1 Description

An instant message connection contains queues for inbound instant messages. A presentity can update the contact status, contact priority, and note for a contact method associated with a connection.
4.4.18.2 Class Definition

The InstantMessageConnection class has attribute values:

- **localNamespace**
  Value: icom_msg

- **localName**
  Value: InstantMessageConnection

- **extendsFrom**
  Value: icom_core:Entity

- **stereotype**
  Value: primary

- **description**
  Value: An instant message connection contains queues for inbound instant messages.

**propertyDefinitions**

The values for this attribute are defined in Section 4.4.18.3.

4.4.18.3 Property Definitions

The InstantMessageConnection class inherits property definitions from super classes.

The InstantMessageConnection class MUST have the property definitions:

- **icom_msg:connectionId**
  Description: An identifier of a connection.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Only

- **icom_msg:selfAddress**
  Description: Address of a presentity who opens a connection.
  Required: True
  Inherited: False
  Property Type: IRI
  Cardinality: Single
  Updatability: On Create
icom_msg: selfResourceName

- Description: Resource name associated with a connection.
- Required: True
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: On Create

icom_msg: inboundInstantMessage

- Description: Inbound instant messages.
- Required: False
- Inherited: False
- Property Type: icom_msg: InstantMessage
- Cardinality: Multi
- Updatability: Read Only

icom_presence: contactStatus

- Description: Reachability status to be propagated to an associated contact method in presence.
- Required: False
- Inherited: False
- Property Type: icom_presence: ContactReachabilityStatus
- Cardinality: Single
- Updatability: Write Only

icom_presence: contactPriority

- Description: Priority to be propagated to an associated contact method in presence.
- Required: False
- Inherited: False
- Property Type: Integer
- Cardinality: Single
- Updatability: Write Only

icom_presence: note

- Description: Note to be propagated to an associated contact method in presence.
- Required: False
- Inherited: False
- Property Type: String
- Cardinality: Single
- Updatability: Write Only
4.5 Presence Module

4.5.1 Presence

4.5.1.1 Description

A presence describes the contact methods and activities of a presentity. It provides a list of contact methods describing how to contact a presentity. A viewer may choose any one of the contact methods based on circumstances. It includes a list of activities describing what a presentity is doing.

4.5.1.2 Class Definition

The Presence class has attribute values:

- **localNamespace**
  - Value: icom_presence

- **localName**
  - Value: Presence

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: primary

- **description**
  - Value: A presence describes the contact methods and activities of a presentity.

4.5.1.3 Property Definitions

The Presence class inherits property definitions from super classes. The Presence class MUST have the property definitions:

- **icom_core:lastModificationDate**
  - Description: Last modification date and time of information in a presence.
  - Required: False
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
  - Updatability: Read Only
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core:location</td>
<td>Location of a presentity.</td>
<td>False</td>
<td>False</td>
<td>icom_core:Location</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_presence:EditMode</td>
<td>Indicates a mode which determines whether a presence is editable.</td>
<td>False</td>
<td>False</td>
<td>icom_presence:PresenceEditMode</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_presence:contactMethod</td>
<td>A collection of contact methods describing how to contact a presentity. A viewer may choose any one of the contact methods based on circumstances.</td>
<td>False</td>
<td>False</td>
<td>icom_presence:ContactMethod</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_presence:activity</td>
<td>A collection of activities describing what a presentity is doing.</td>
<td>False</td>
<td>False</td>
<td>icom_presence:Activity</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

The Presence class MAY include additional property definitions which are implementation-defined.
4.5.2 PresenceEditMode

4.5.2.1 Description
A presence edit mode is a mode that indicates whether a presence is editable.

4.5.2.2 Class Definition
The PresenceEditMode class is a mixin class which defines a mode that indicates whether a presence is editable.

The PresenceEditMode class has attribute values:

- **localNamespace**
  Value: icom_presence

- **localName**
  Value: PresenceEditMode

- **extendsFrom**
  Value:
**stereotype**
Value: mixin

description
Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a presence is editable.

**propertyDefinitions**
The values for this attribute are defined in Section 4.5.2.3.

### 4.5.2.3 Property Definitions
The PresenceEditMode class MAY include additional property definitions which are implementation-defined.

### 4.5.3 PresenceEditModeEnum
The PresenceEditModeEnum class is an enum class that enumerates the instances each of which expresses a mode that indicates whether a presence is editable.

The PresenceEditModeEnum class has attribute values:

**localNamespace**
Value: icom_presence

**localName**
Value: PresenceEditModeEnum

**extendsFrom**
Value: icom_presence:PresenceEditMode

**stereotype**
Value: primary

**isEnumeration**
Value: TRUE

**description**
Value: A mode that indicates whether a presence is editable.

**instances**
Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>

ICOM defines two presence editable modes:
• **icom\_presence:PresentityCopy** a presence is a copy belonging to a presentity who may update the properties such as activities.

• **icom\_presence:ViewerCopy** a presence is a copy visible to a subscriber who may not update the properties.

### 4.5.4 ContactMethod

#### 4.5.4.1 Description

A contact method object describes reachability circumstances of a presentity.

#### 4.5.4.2 Class Definition

The ContactMethod class has attribute values:

- **localNamespace**
  - Value: icom\_presence

- **localName**
  - Value: ContactMethod

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: A contact method object describes reachability circumstances of a presentity.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.5.4.3

### 4.5.4.3 Property Definitions

The ContactMethod class MUST have the property definitions:

- **icom\_core:creationDate**
  - Description: Creation date and time of information in a contact method.
  - Required: False
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
  - Updatability: Read Only
icom_core:lastModificationDate
   Description: Last modification date and time of information in a contact method.
   Required: False
   Inherited: False
   Property Type: DateTime
   Cardinality: Single
   Updatability: Read Only

icom_presence:activeConnectionId
   Description: A list of active connection ids of a presentity.
   Required: False
   Inherited: False
   Property Type: String
   Cardinality: Multi
   Updatability: Read Only

icom_presence:contactEndpoint
   Description: A list of endpoints or IRIs for contacting a presentity.
   Required: False
   Inherited: False
   Property Type: String
   Cardinality: Multi
   Updatability: Read Only

icom_presence:contactPriority
   Description: Priority of a contact method relative to other contact methods in a presence.
   Required: False
   Inherited: False
   Property Type: Integer
   Cardinality: Single
   Updatability: Read Only

icom_presence:contactStatus
   Description: Status of a contact method in a presence.
   Required: False
   Inherited: False
   Property Type: icom_presence:ContactReachabilityStatus
   Cardinality: Single
   Updatability: Read Only
icom_presence:note

Description: A note about contacting a presentity.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Only

4.5.5 ContactReachabilityStatus

4.5.5.1 Description
A contact reachability status is a status of a contact method.

4.5.5.2 Class Definition
The ContactReachabilityStatus class is a mixin class which defines a status of a contact method.
The ContactReachabilityStatus class has attribute values:

- **localNamespace**
  - Value: icom_presence

- **localName**
  - Value: ContactReachabilityStatus

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

- **description**
  - Value: ContactReachabilityStatus is a mixin class which defines a status of a contact method.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.5.5.3.

4.5.5.3 Property Definitions
The ContactReachabilityStatus class MAY include additional property definitions which are implementation-defined.

4.5.6 ContactReachabilityStatusEnum
The ContactReachabilityStatusEnum class is an enum class that enumerates the instances each of which expresses a reachability status of a contact method.
The ContactReachabilityStatusEnum class has attribute values:

- **localNamespace**
  - Value: icom_presence

- **localName**
  - Value: ContactReachabilityStatusEnum

- **extendsFrom**
  - Value: icom_presence:ContactReachabilityStatus

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A reachability status of a contact method.

- **instances**

ICOM defines six reachability status:

- **icom_presence:Reachable** a presentity is reachable through a contact method.
- **icom_presence:NotReachable** a presentity is not reachable through a contact method.
- **icom_presence:Chatty** a presentity is chatty.
- **icom_presence:Away** a presentity is away.
- **icom_presence:ExtendedAway** a presentity is away for an extended period.
- **icom_presence:DoNotDisturb** a presentity prefers not to be disturbed.
Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.
4.5.7 Activity

4.5.7.1 Description
An activity object describes what a presentity is doing.

4.5.7.2 Class Definition
The Activity class has attribute values:

- `localNamespace`
  - Value: icom_presence

- `localName`
  - Value: Activity

- `extendsFrom`
  - Value: 

- `stereotype`
  - Value: primary

- `description`
  - Value: An activity object describes what a presentity is doing.

- `propertyDefinitions`
  - The values for this attribute are defined in Section 4.5.7.3.

4.5.7.3 Property Definitions
The Activity class MUST have the property definitions:

- `icom_core:startDate`
  - Description: Start date and time of an activity.
  - Required: True
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
  - Updatability: Read Write

- `icom_core:endDate`
  - Description: End date and time of an activity.
  - Required: True
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
### 4.5.8 ActivityType

#### 4.5.8.1 Description
An activity type is a vocabulary of activities for rich presence information model.

#### 4.5.8.2 Class Definition
The ActivityType class is a mixin class which defines an activity.

The ActivityType class has attribute values:

- **localNamespace**
  - Value: `icom_presence`

- **localName**
  - Value: `ActivityType`
extendsFrom
    Value:

stereotype
    Value: mixin

description
    Value: ActivityType is a mixin class which defines a type of activity.

propertyDefinitions
    The values for this attribute are defined in Section 4.5.8.3.

4.5.8.3 Property Definitions
The ActivityType class MAY include additional property definitions which are implementation-defined.

4.5.9 ActivityTypeEnum
The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a type of activity.
The ActivityTypeEnum class has attribute values:

localNamespace
    Value: icom_presence

localName
    Value: ActivityTypeEnum

extendsFrom
    Value: icom_presence:ActivityType

stereotype
    Value: primary

isEnumeration
    Value: TRUE

description
    Value: A type of activity.

instances
ICOM defines eleven activity types:

- **icom_presence:OnThePhone** a presentity is on the phone.
- **icom_presence:Conference** a presentity is in a conference.
- **icom_presence:Meeting** a presentity is in a meeting.
- **icom_presence:Travel** a presentity is traveling.
- **icom_presence:Steering** a presentity is steering a vehicle.
- **icom_presence:Meal** a presentity is having a meal.
- **icom_presence:OutOfOffice** a presentity is out of office.
- **icom_presence:Holiday** a presentity is on holiday.
- **icom_presence:Vacation** a presentity is on vacation.
- **icom_presence:OutOfContact** a presentity is out of contact.
- **icom_presence:OtherActivityType** a presentity is involved in an unspecified activity.

### 4.6 Address Book Module

#### 4.6.1 AddressBook

##### 4.6.1.1 Description

An address book is a folder that contains sub-address books and addressable contacts.

##### 4.6.1.2 Class Definition

The AddressBook class has attribute values:

- **localNamespace**
  - Value: `icom_card`

- **localName**
  - Value: `AddressBook`

- **extendsFrom**
  - Value: `icom_core:Folder`

- **stereotype**
  - Value: `primary`

- **description**
  - Value: An address book is a folder that contains sub-address books and addressable contacts.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.6.1.3.
4.6.1.3 Property Definitions

The AddressBook class inherits property definitions from super classes. The AddressBook class MUST have the property definitions:

icom_card:addressBook

- Description: Sub-address books in an address book.
- Required: False
- Inherited: False
- Property Type: icom_card:AddressBook
- Cardinality: Multi
- Updatability: Read Only

icom_card:contact

- Description: Contacts in an address book.
- Required: False
- Inherited: False
- Property Type: icom_card:PersonContact
- Cardinality: Multi
- Updatability: Read Only

The AddressBook class MAY include additional property definitions which are implementation-defined.

Figure 34: Address Book Class Diagram.

4.6.2 PersonContact

4.6.2.1 Description

A person contact is an artifact that contains address information about a person.
4.6.2.2 Class Definition

The PersonContact class has attribute values:

localNamespace

Value: icom_card

localName

Value: PersonContact

extendsFrom

Value: icom_core:Artifact, icom_core:Addressable

stereotype

Value: primary

description

Value: A person contact is an artifact that contains address information about a person.

propertyDefinitions

The values for this attribute are defined in Section 4.6.2.3.

4.6.2.3 Property Definitions

The PersonContact class inherits property definitions from super classes.

The PersonContact class MUST have the property definitions:

icom_core:timeZone

Description: Time zone of a person.

Required: False

Inherited: False

Property Type: icom_core:TimeZone

Cardinality: Single

Updatability: Read Write

icom_core:givenName

Description: Given name of a person.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write
icom_core:middleName
Description: Middle name of a person. Can include multiple names concatenated.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:familyName
Description: Family name of a person.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:prefix
Description: Prefix of a person's name.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:suffix
Description: Suffix of a person's name.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:nickname
Description: Nickname of a person.
Required: False
Inherited: False
Property Type: String
Cardinality: Multi
Updatability: Read Write

icom_core:jobTitle
Description: Job title of a person.
icom_core:department

Description: A person’s affiliated department.

Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:officeLocation

Description: Location of a person’s department.

Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:company

Description: A person’s affiliated company.

Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:profession

Description: A person’s profession.

Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_content:attachment

Description: One or more content attachments in a contact.

Required: False
Inherited: False
Property Type: icom_content:AttachedItem
Cardinality: Multi
Updatability: Read Write

icom_card:bookmark
Description: A person which is bookmarked by a contact.
Required: False
Inherited: False
Property Type: icom_core:Person
Cardinality: Single
Updatability: On Create

The PersonContact class MAY include additional property definitions which are implementation-defined.
Figure 35: Person Contact Class Diagram.

4.7 Calendar Module

4.7.1 Calendar

4.7.1.1 Description
A calendar contains time management artifacts that include occurrences and occurrence series.

4.7.1.2 Class Definition
The Calendar class has attribute values:
localNamespace
Value: icom_cal

localName
Value: Calendar

extendsFrom
Value: icom_core:Folder

stereotype
Value: primary

description
Value: A calendar contains time management artifacts that include occurrences and occurrence series.

propertyDefinitions
The values for this attribute are defined in 4.7.1.3.

4.7.1.3 Property Definitions
The Calendar class inherits property definitions from super classes.
The Calendar class MUST have the property definitions:

icom_core:timeZone
Description: Time zone setting for a calendar.
Required: True
Inherited: False
Property Type: icom_core:TimeZone
Cardinality: Single
Updatability: Read Write

icom_core:element
Description: Elements of a calendar.
Required: False
Inherited: True
Property Type: icom_cal:Occurrence
Cardinality: Multi
Updatability: Read Only

icom_cal:recurrence
Description: Occurrence series of a calendar.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceSeries
Cardinality: Multi
Updatability: Read Only

Figure 36: Calendar Class Diagram.

4.7.2 OccurrenceSeries

4.7.2.1 Description
An occurrence series represents a series of occurrences associated with the same calendar event.

4.7.2.2 Class Definition
The OccurrenceSeries class has attribute values:

localNamespace
Value: icom_cal

localName
Value: OccurrenceSeries

extendsFrom
Value: icom_core:Artifact

 stereotype
Value: primary

description
Value: An occurrence series represents a series of occurrences associated with the same calendar event.

propertyDefinitions
The values for this attribute are defined in 4.7.2.3.
### 4.7.2.3 Property Definitions

The OccurrenceSeries class inherits property definitions from super classes. The OccurrenceSeries class MUST have the property definitions:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core:location</td>
<td>Location of an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>icom_core:Location</td>
<td>Organizer of an occurrence series.</td>
<td>True</td>
<td>False</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icom_core:organizer</td>
<td>Participants in an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>icom_cal:OccurrenceParticipant</td>
<td></td>
<td></td>
<td></td>
<td>Multi</td>
<td>On Create</td>
</tr>
<tr>
<td>icom_core:priority</td>
<td>Priority for an attendee of an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>icom_core:Priority</td>
<td></td>
<td></td>
<td></td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icom_content:attachment</td>
<td>One or more content attachments in an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>icom_content:AttachedItem</td>
<td></td>
<td></td>
<td></td>
<td>Multi</td>
<td></td>
</tr>
</tbody>
</table>
icom_cal:recurrenceStartDate

Description: Start date and time of an occurrence series.
Required: True
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: On Create

icom_cal:recurrenceStartDateResolution

Description: Resolution of start date and time of an occurrence series.
Required: True
Inherited: False
Property Type: icom_core:DateTimeResolution
Cardinality: Single
Updatability: On Create

icom_cal:duration

Description: Duration of each occurrence in an occurrence series.
Required: True
Inherited: False
Property Type: Duration
Cardinality: Single
Updatability: On Create

icom_cal:recurrenceRule

Description: A recurrence rule of an occurrence series.
Required: True
Inherited: False
Property Type: String
Cardinality: Single
Updatability: On Create

icom_cal:occurrenceStatus

Description: Status of an occurrence series.
Required: True
Inherited: False
Property Type: icom_cal:OccurrenceStatus
Cardinality: Single
Updatability: Read Write
icom_cal:occurrenceType

Description: Type of an occurrence series.
Required: True
Inherited: False
Property Type: icom_cal:OccurrenceType
Cardinality: Single
Updatability: Read Write

icom_cal:editMode

Description: Indicates a mode which determines whether an occurrence series is editable.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceEditMode
Cardinality: Single
Updatability: Read Only

icom_cal:occurrence

Description: Occurrences in an occurrence series.
Required: False
Inherited: False
Property Type: icom_cal:Occurrence
Cardinality: Multi
Updatability: Read Only

icom_cal:attendee

Description: An attendee of an occurrence series.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Single
Updatability: Read Only

icom_cal:attendeeParticipantStatus

Description: Participation status for an attendee of an occurrence series.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceParticipantStatus
Cardinality: Single
Updatability: Read Write
icom_cal:transparency

Description: Participant transparency for an attendee of an occurrence series.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceParticipantTransparency
Cardinality: Single
Updatability: Read Write

icom_cal:attendeeProperty

Description: Extensible properties for an attendee of an occurrence series.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

icom_conf:conference

Description: One or more conferences for an occurrence series.
Required: False
Inherited: False
Property Type: icom_conf:Conference
Cardinality: Multi
Updatability: Read Write
4.7.3 Occurrence

4.7.3.1 Description
An occurrence represents an event in a calendar.

4.7.3.2 Class Definition
The Occurrence class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: Occurrence
extendsFrom
   Value: icom_core:Artifact

stereotype
   Value: primary

description
   Value: An occurrence represents an event in a calendar.

propertyDefinitions
   The values for this attribute are defined in 4.7.3.3.

4.7.3.3 Property Definitions
The Occurrence class inherits property definitions from super classes.
The Occurrence class MUST have the property definitions:

icom_core:location
   Description: Location of an occurrence.
   Required: False
   Inherited: False
   Property Type: icom_core:Location
   Cardinality: Single
   Updatability: Read Write

icom_core:organizer
   Description: Organizer of an occurrence.
   Required: True
   Inherited: False
   Property Type: icom_core:Participant
   Cardinality: Single
   Updatability: On Create

icom_core:participant
   Description: Participants of an occurrence.
   Required: False
   Inherited: False
   Property Type: icom_cal:OccurrenceParticipant
   Cardinality: Multi
   Updatability: Read Write
### icom_core:priority

**Description:** Priority for an attendee of an occurrence.

**Required:** False

**Inherited:** False

**Property Type:** icom_core:Priority

**Cardinality:** Single

**Updatability:** Read Write

### icom_core:startDate

**Description:** Start date and time of an occurrence.

**Required:** True

**Inherited:** False

**Property Type:** DateTime

**Cardinality:** Single

**Updatability:** On Create

### icom_core:startDateResolution

**Description:** Resolution of start date and time of an occurrence.

**Required:** True

**Inherited:** False

**Property Type:** icom_core:DateTimeResolution

**Cardinality:** Single

**Updatability:** On Create

### icom_core:endDate

**Description:** End date and time of an occurrence.

**Required:** True

**Inherited:** False

**Property Type:** DateTime

**Cardinality:** Single

**Updatability:** On Create

### icom_core:endDateResolution

**Description:** Resolution of end date and time of an occurrence.

**Required:** True

**Inherited:** False

**Property Type:** icom_core:DateTimeResolution

**Cardinality:** Single

**Updatability:** On Create

### icom_content:attachment

**Description:** One or more content attachments in an occurrence.
Required: False
Inherited: False
Property Type: icom_content:AttachedItem
Cardinality: Multi
Updatability: Read Write

icom_cal:occurrenceSeries
Description: An occurrence is part of this occurrence series.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceSeries
Cardinality: Single
Updatability: Read Only

icom_cal:fromRecurringOccurrenceSeries
Description: Occurrence is part of a recurring occurrence series.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Only

icom_cal:exceptionToOccurrenceSeries
Description: Occurrence is an exception to an occurrence series.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Only

icom_cal:occurrenceStatus
Description: Status of an occurrence.
Required: True
Inherited: False
Property Type: icom_cal:OccurrenceStatus
Cardinality: Single
Updatability: Read Write

icom_cal:occurrenceType
Description: Type of an occurrence.
Required: True
Inherited: False
Property Type: `icom_cal:OccurrenceType`
Cardinality: Single
Updatability: Read Write

**icom_cal:editMode**

Description: Indicates a mode which determines whether an occurrence is editable.
Required: False
Inherited: False
Property Type: `icom_cal:OccurrenceEditMode`
Cardinality: Single
Updatability: Read Only

**icom_cal:attendee**

Description: An attendee of an occurrence.
Required: False
Inherited: False
Property Type: `icom_core:Participant`
Cardinality: Single
Updatability: Read Only

**icom_cal:attendeeParticipantStatus**

Description: Status for an attendee of an occurrence.
Required: False
Inherited: False
Property Type: `icom_cal:OccurrenceParticipantStatus`
Cardinality: Single
Updatability: Read Write

**icom_cal:transparency**

Description: Transparency for an attendee of an occurrence.
Required: False
Inherited: False
Property Type: `icom_cal:OccurrenceParticipantTransparency`
Cardinality: Single
Updatability: Read Write

**icom_cal:attendeeProperty**

Description: Extensible properties for an attendee of an occurrence.
Required: False
Inherited: False
Property Type: `icom_meta:Property`
Cardinality: Multi
6405 Updatability: Read Write
6406
6407 **icom_conf:conference**
6408 Description: One or more conferences for an occurrence.
6409 Required: False
6410 Inherited: False
6411 Property Type: icom_conf:Conference
6412 Cardinality: Multi
6413 Updatability: Read Write
6414
6415
6416 Figure 38: Occurrence Class Diagram.
6417
4.7.4 OccurrenceStatus

4.7.4.1 Description
An occurrence status is a status of a calendar occurrence.

4.7.4.2 Class Definition
The OccurrenceStatus class is a mixin class which defines status of a calendar occurrence.

The OccurrenceStatus class has attribute values:

- `localNamespace`: Value: `icom_cal`
- `localName`: Value: `OccurrenceStatus`
- `extendsFrom`
  Value:
- `stereotype`: Value: `mixin`
- `description`: Value: `OccurrenceStatus is a mixin class which defines status of a calendar occurrence.`
- `propertyDefinitions`

4.7.4.3 Property Definitions
The OccurrenceStatus class MAY include additional property definitions which are implementation-defined.

4.7.5 OccurrenceStatusEnum

The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which expresses a status of an occurrence or occurrence series.

The OccurrenceStatusEnum class has attribute values:

- `localNamespace`: Value: `icom_cal`
- `localName`: Value: `OccurrenceStatusEnum`
ICOM defines three occurrence status:

- **icom_cal:Cancelled** an occurrence or occurrence series is cancelled.
- **icom_cal:Tentative** an occurrence or occurrence series is tentative.
- **icom_cal:Confirmed** an occurrence or occurrence series is confirmed.

### 4.7.6 OccurrenceType

#### 4.7.6.1 Description

An occurrence type is a category of calendar occurrences.

#### 4.7.6.2 Class Definition

The OccurrenceType class is a mixin class which defines a type of occurrence.

The OccurrenceType class has attribute values:

- **localNamespace**
  - Value: icom_cal
- **localName**
  - Value: OccurrenceType
- **extendsFrom**
  - Value: 
- **stereotype**
  - Value: mixin
description
  Value: OccurrenceType is a mixin class which defines a type of occurrence.

propertyDefinitions
  The values for this attribute are defined in Section 4.7.6.3.

4.7.6.3 Property Definitions
The OccurrenceType class MAY include additional property definitions which are implementation-defined.

4.7.7 OccurrenceTypeEnum
The OccurrenceTypeEnum class is an enum class that enumerates the instances each of which expresses a type of an occurrence or occurrence series.
The OccurrenceTypeEnum class has attribute values:

  localNamespace
    Value: icom_cal

  localName
    Value: OccurrenceTypeEnum

  extendsFrom
    Value: icom_cal:OccurrenceType

  stereotype
    Value: primary

  isEnumeration
    Value: TRUE

  description
    Value: Type of an occurrence or occurrence series.

  instances
    Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry, icom_cal:OtherOccurrenceType>

ICOM defines five occurrence types:

- icom_cal:Meeting an occurrence or occurrence series is a meeting.
- icom_cal:DayEvent an occurrence or occurrence series is a day event.
- icom_cal:Holiday an occurrence or occurrence series is a holiday.
- icom_cal:JournalEntry an occurrence or occurrence series is a journal entry.
- icom_cal:OtherOccurrenceType an occurrence or occurrence series is of other type.
4.7.8 OccurrenceParticipant

4.7.8.1 Description
An occurrence participant object is a participant object that contains an occurrence participant status.

4.7.8.2 Class Definition
The OccurrenceParticipant class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: OccurrenceParticipant

- **extendsFrom**
  - Value: icom_core:Participant

- **stereotype**
  - Value: primary

- **description**
  - Value: An occurrence participant object is a participant object that contains an occurrence participant status.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.7.8.3.

4.7.8.3 Property Definitions
The OccurrenceParticipant class inherits property definitions from super classes.
The OccurrenceParticipant class MUST have the property definition:

- **icom_cal:participantStatus**
  - Description: Status of an occurrence participant.
  - Required: False
  - Inherited: False
  - Property Type: icom_cal:OccurrenceParticipantStatus
  - Cardinality: Single
  - Updatability: Read Write
4.7.9 OccurrenceParticipantStatus

4.7.9.1 Description
An occurrence participant status is a participant’s response status for an occurrence or occurrence series.

4.7.9.2 Class Definition
The OccurrenceParticipantStatus class is a mixin class which defines a participant’s response status for an occurrence or occurrence series.

The OccurrenceParticipantStatus class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: OccurrenceParticipantStatus

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

- **description**
  - Value: OccurrenceParticipantStatus is a mixin class which defines a participant’s response status for an occurrence or occurrence series.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.7.9.3.

4.7.9.3 Property Definitions
The OccurrenceParticipantStatus class MAY include additional property definitions which are implementation-defined.

4.7.10 OccurrenceParticipantStatusEnum
The OccurrenceParticipantStatusEnum class is an enum class that enumerates the instances each of which expresses a participant’s response status for an occurrence or occurrence series.

The OccurrenceParticipantStatusEnum class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: OccurrenceParticipantStatusEnum
extendsFrom
    Value: icom_cal:OccurrenceParticipantStatus

stereotype
    Value: primary

isEnumeration
    Value: TRUE

description
    Value: Participant's response status for an occurrence or occurrence series.

instances
    Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>

ICOM defines four occurrence participant's status:
- icom_cal:NeedsAction an attendee needs to act on an occurrence or occurrence series.
- icom_cal:Accepted an attendee accepted an occurrence or occurrence series.
- icom_cal:Declined an attendee declined an occurrence or occurrence series.
- icom_cal:Tentative an attendee is tentative about attending an occurrence or occurrence series.

4.7.11 OccurrenceParticipantTransparency

4.7.11.1 Description
An occurrence participant transparency is visibility of an occurrence or occurrence series in a participant's calendar or free busy.

4.7.11.2 Class Definition
The OccurrenceParticipantTransparency class is a mixin class which defines visibility of an occurrence or occurrence series in a participant's calendar or free busy.
The OccurrenceParticipantTransparency class has attribute values:

    localNamespace
        Value: icom_cal

    localName
        Value: OccurrenceParticipantTransparency

    extendsFrom
        Value:
**stereotype**
Value: mixin

**description**
Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an occurrence or occurrence series in a participant’s calendar or free busy.

**propertyDefinitions**
The values for this attribute are defined in Section 4.7.11.3.

### 4.7.11.3 Property Definitions

The OccurrenceParticipantTransparency class MAY include additional property definitions which are implementation-defined.

### 4.7.12 OccurrenceParticipantTransparencyEnum

The OccurrenceParticipantTransparencyEnum class is an enum class that enumerates the instances each of which expresses an occurrence or occurrence series transparency in a participant’s calendar or free busy.

The OccurrenceParticipantTransparencyEnum class has attribute values:

**localNamespace**
Value: icom_cal

**localName**
Value: OccurrenceParticipantTransparencyEnum

**extendsFrom**
Value: icom_cal:OccurrenceParticipantTransparency

**stereotype**
Value: primary

**isEnumeration**
Value: TRUE

**description**
Value: Occurrence or occurrence series transparency in a participant’s calendar or free busy.

**instances**
ICOM defines five participant transparencies:

- **icom_cal:Opaque** an occurrence or occurrence series is opaque in a participant’s calendar or free busy.
- **icom_cal:Transparent** an occurrence or occurrence series is transparent in a participant’s calendar or free busy.
- **icom_cal:Tentative** an occurrence or occurrence series has a tentative transparency in a participant’s calendar or free busy.
- **icom_cal:OutOfOffice** an occurrence or occurrence series has out of office transparency in a participant’s calendar or free busy.
- **icom_cal:DefaultTransparency** an occurrence or occurrence series has default transparency in a participant’s calendar or free busy.

### 4.7.13 OccurrenceEditMode

#### 4.7.13.1 Description

An occurrence edit mode is a mode that indicates whether an occurrence or occurrence series is editable.

#### 4.7.13.2 Class Definition

The OccurrenceEditMode class is a mixin class which defines a mode that indicates whether an occurrence or occurrence series is editable.

The OccurrenceEditMode class has attribute values:

- **localNamespace**
  Value: icom_cal

- **localName**
  Value: OccurrenceEditMode

- **extendsFrom**
  Value:

- **stereotype**
  Value: mixin

- **description**
  Value: OccurrenceEditMode is a mixin class which defines a mode that indicates whether an occurrence or occurrence series is editable.

- **propertyDefinitions**
  The values for this attribute are defined in Section 4.7.13.3.

#### 4.7.13.3 Property Definitions

The OccurrenceEditMode class MAY include additional property definitions which are implementation-defined.
4.7.14 OccurrenceEditModeEnum

The OccurrenceEditModeEnum class is an enum class that enumerates the instances each of which expresses a mode that indicates whether an occurrence or occurrence series is editable.

The OccurrenceEditModeEnum class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: OccurrenceEditMode

- **extendsFrom**
  - Value: icom_cal:OccurrenceEditMode

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A mode that indicates whether an occurrence or occurrence series is editable.

- **instances**
  - Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

ICOM defines two occurrence editable modes:

- **icom_cal:OrganizerCopy** an occurrence or occurrence series is a copy created by an organizer who may update the properties such as occurrence type, occurrence status, etc.
- **icom_cal:AttendeeCopy** an occurrence or occurrence series is a copy delivered to an attendee who may only update the attendee properties such as priority, transparency, etc.

4.8 Free Busy Module

4.8.1 FreeBusy

4.8.1.1 Description

A free busy object specifies the free time and busy time intervals of one or more participants.
4.8.1.2 Class Definition

The FreeBusy class has attribute values:

- **localNamespace**
  - Value: `icom_cal`

- **localName**
  - Value: FreeBusy

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: primary

- **description**
  - Value: A free busy object specifies the free time and busy time intervals of one or more participants.

4.8.1.3 Property Definitions

The FreeBusy class MUST have the property definitions:

- **icom_core:participant**
  - Description: A list of participants whose free busy intervals are included.
  - Required: False
  - Inherited: False
  - Property Type: `icom_core:Participant`
  - Cardinality: Multi
  - Updatability: Read Only

- **icom_core:creationDate**
  - Description: Creation date and time of a free busy object.
  - Required: False
  - Inherited: False
  - Property Type: `DateTime`
  - Cardinality: Single
  - Updatability: Read Only

- **icom_core:startDate**
  - Description: Start date and time of a list of free busy intervals.
4.8.2 FreeBusyInterval

4.8.2.1 Description

A free busy interval specifies an interval of free or busy time.

If a free busy type is icom_cal:Free, then a time interval is free for scheduling.

If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been scheduled for the interval.

4.8.2.2 Class Definition

The FreeBusyInterval class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: FreeBusyInterval

- **extendsFrom**
  - Value:
skeleton

value: primary

description

value: A free busy interval object specifies an interval of free or busy time.

propertyDefinitions

The values for this attribute are defined in Section 4.8.2.3

4.8.2.3 Property Definitions

The FreeBusyInterval class MUST have the property definitions:

icom_core:startDate

description: Start date and time of a free busy interval.

required: False

inherited: False

property type: DateTime

cardinality: Single

updatability: Read Only

icom_core:endDate

description: End date and time of a free busy interval.

required: False

inherited: False

property type: DateTime

cardinality: Single

updatability: Read Only

icom_cal:freeBusyType

description: A type of free busy interval.

required: False

inherited: False

property type: icom_cal:FreeBusyType

cardinality: Single

updatability: Read Only
4.8.3 FreeBusyType

4.8.3.1 Description

A free busy type classifies a time interval as free, busy, or other.

4.8.3.2 Class Definition

The FreeBusyType class is a mixin class which defines different types to classify a time interval.

The FreeBusyType class has attribute values:

- `localNamespace`
  - Value: icom_cal

- `localName`
  - Value: FreeBusyType

- `extendsFrom`
  - Value: 

- `stereotype`
  - Value: mixin

- `description`
  - Value: FreeBusyType is a mixin class which defines different types to classify a time interval.

Figure 39: Free Busy Class Diagram.
The values for this attribute are defined in Section 4.8.3.3.

4.8.3.3 Property Definitions

The FreeBusyType class MAY include additional property definitions which are implementation-defined.

4.8.4 FreeBusyTypeEnum

The FreeBusyTypeEnum class is an enum class that enumerates the instances each of which expresses a type of free busy interval.

The FreeBusyTypeEnum class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: FreeBusyTypeEnum

- **extendsFrom**
  - Value: icom_cal:FreeBusyType

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A type of free busy interval.

- **instances**
  - Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours, icom_cal:OutOfOffice, icom_cal:OtherFreeBusyType>

ICOM defines six free busy types:

- **icom_cal:Free** a free busy interval is free.
- **icom_cal:Busy** a free busy interval is busy.
- **icom_cal:Tentative** a free busy interval is tentative.
- **icom_cal:OutsideAvailableHours** a free busy interval is outside available hours.
- **icom_cal:OutOfOffice** a free busy interval is within out of office hours.
- **icom_cal:OtherFreeBusyType** a free busy interval is of other type.
4.9 Task List Module

4.9.1 TaskList

4.9.1.1 Description
A task list contains task management artifacts.

4.9.1.2 Class Definition
The TaskList class has attribute values:

- **localNamespace**
  - Value: icom_task

- **localName**
  - Value: TaskList

- **extendsFrom**
  - Value: icom_core:Folder

- **stereotype**
  - Value: primary

- **description**
  - Value: A task list contains task management artifacts.

4.9.1.3 Property Definitions
The TaskList class inherits property definitions from super classes.

The TaskList class MUST have the property definitions:

- **icom_core:timeZone**
  - Description: Time zone of a task list.
  - Required: True
  - Inherited: False
  - Property Type: icom_core:TimeZone
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:element**
  - Description: Elements of a task list.
  - Required: False
Inherited: True
Property Type: icom_task:Task
Cardinality: Multi
Updatability: Read Only

Figure 40: Task List Class Diagram.

4.9.2 Task

4.9.2.1 Description
A task is an artifact that represents a task to do or a task assignment in a task list.

4.9.2.2 Class Definition
The Task class has attribute values:

localNamespace
  Value: icom_task

localName
  Value: Task

extendsFrom
  Value: icom_core:Artifact

stereotype
  Value: primary

description
  Value: A task is an artifact that represents a task to do or a task assignment in a task list.
propertyDefinitions

The values for this attribute are defined in 4.9.2.3.

### 4.9.2.3 Property Definitions

The Task class inherits property definitions from super classes.

The Task class MUST have the property definitions:

icom_core:location

- Description: Location of a task.
- Required: False
- Inherited: False
- Property Type: icom_core:Location
- Cardinality: Single
- Updatability: Read Write

icom_core:organizer

- Description: Organizer of a task.
- Required: True
- Inherited: False
- Property Type: icom_core:Participant
- Cardinality: Single
- Updatability: On Create

icom_core:priority

- Description: Priority of a task.
- Required: False
- Inherited: False
- Property Type: icom_core:Priority
- Cardinality: Single
- Updatability: Read Write

icom_core:startDate

- Description: Start date and time of a task.
- Required: True
- Inherited: False
- Property Type: DateTime
- Cardinality: Single
- Updatability: On Create

icom_core:startDateResolution

- Description: Resolution of start date and time of a task.
- Required: True
Inherited: False

Property Type: icom_core:DateTimeResolution
Cardinality: Single
Updatability: On Create

icom_content:attachment

Description: One or more content attachments in a task.
Required: False
Inherited: False
Property Type: icom_content:AttachedItem
Cardinality: Multi
Updatability: Read Write

icom_task:dueDate

Description: Due date and time of a task.
Required: True
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: On Create

icom_task:dueDateResolution

Description: Resolution of due date and time of a task.
Required: True
Inherited: False
Property Type: icom_core:DateTimeResolution
Cardinality: Single
Updatability: On Create

icom_task:editMode

Description: Indicates a mode which determines whether a task is editable.
Required: False
Inherited: False
Property Type: icom_task:TaskEditMode
Cardinality: Single
Updatability: Read Only

icom_task:taskStatus

Description: Status of a task.
Required: True
Inherited: False
Property Type: icom_task:TaskStatus
**icom_task:assignee**

- **Description:** An assignee of a task.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:Participant
- **Cardinality:** Single
- **Updatability:** Read Only

**icom_task:participantStatus**

- **Description:** Participation status of a task.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_task:TaskParticipantStatus
- **Cardinality:** Single
- **Updatability:** Read Write

**icom_task:completionDate**

- **Description:** Completion date and time of a task.
- **Required:** False
- **Inherited:** False
- **Property Type:** DateTime
- **Cardinality:** Single
- **Updatability:** Read Write

**icom_task:completionDateResolution**

- **Description:** Resolution of completion date and time of a task.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:DateTimeResolution
- **Cardinality:** Single
- **Updatability:** Read Write

**icom_task:percentComplete**

- **Description:** Percentage of task completed.
- **Required:** False
- **Inherited:** False
- **Property Type:** Integer
- **Cardinality:** Single
- **Updatability:** Read Write
icom_task:assigneeProperty

Description: Extensible properties for an assignee of a task.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

---

Figure 41: Task Class Diagram.
4.9.3 TaskStatus

4.9.3.1 Description
A task status is a status of a task.

4.9.3.2 Class Definition
The TaskStatus class is a mixin class which defines status of a task.
The TaskStatus class has attribute values:

```plaintext
localNamespace
   Value: icom_task

localName
   Value: TaskStatus

extendsFrom
   Value:

stereotype
   Value: mixin

description
   Value: TaskStatus is a mixin class which defines status of a task.

propertyDefinitions
   The values for this attribute are defined in Section 4.9.3.3.
```

4.9.3.3 Property Definitions
The TaskStatus class MAY include additional property definitions which are implementation-defined.

4.9.4 TaskStatusEnum
The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a status of task.
The TaskStatusEnum class has attribute values:

```plaintext
localNamespace
   Value: icom_task

localName
   Value: TaskStatusEnum

extendsFrom
   Value: icom_task:TaskStatus
```
ICOM defines four task statuses:

- **icom_task:NeedsAction** a task needs action.
- **icom_task:InProgress** a task is in progress.
- **icom_task:Completed** a task is completed.
- **icom_task:Cancelled** a task is cancelled.

### 4.9.5 TaskParticipantStatus

#### 4.9.5.1 Description

A task participant status is a participant’s response status for a task assignment.

#### 4.9.5.2 Class Definition

The TaskParticipantStatus class is a mixin class which defines a participant’s response status for a task assignment.

The TaskParticipantStatus class has attribute values:

- **localNamespace**
  - Value: icom_task

- **localName**
  - Value: TaskParticipantStatus

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin
description
   Value: TaskParticipantStatus is a mixin class which defines a participant’s response status for a task assignment.

propertyDefinitions
   The values for this attribute are defined in Section 4.9.5.3.

4.9.5.3 Property Definitions
   The TaskParticipantStatus class MAY include additional property definitions which are implementation-defined.

4.9.6 TaskParticipantStatusEnum
   The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which expresses a participant’s response status for a task.
   The TaskParticipantStatusEnum class has attribute values:

   localNamespace
      Value: icom_task

   localName
      Value: TaskParticipantStatusEnum

   extendsFrom
      Value: icom_task:TaskParticipantStatus

   stereotype
      Value: primary

   isEnumeration
      Value: TRUE

   description
      Value: Participant’s response status for a task.

   instances
      Value: <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined, icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative, icom_task:Deferred>

ICOM defines eight task participant’s status:
   • icom_task:NeedsAction an assignee needs to act on a task.
   • icom_task:Accepted an assignee accepted a task.
   • icom_task:Declined an assignee declined a task.
- **icom_task:InProgress** a task is in progress.
- **icom_task:Completed** a task is completed.
- **icom_task:WaitingOnOther** an assignee is waiting on other.
- **icom_task:Tentative** an assignee is tentative about a task.
- **icom_task:Deferred** an assignee deferred a task.

### 4.9.7 TaskEditMode

#### 4.9.7.1 Description
A task edit mode is a mode that indicates whether a task is editable.

#### 4.9.7.2 Class Definition
The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

The TaskEditMode class has attribute values:

- **localNamespace**
  Value: iom_task

- **localName**
  Value: TaskEditMode

- **extendsFrom**
  Value: 

- **stereotype**
  Value: mixin

- **description**
  Value: TaskEditMode is a mixin class which defines a mode that indicates whether task is editable.

- **propertyDefinitions**
  The values for this attribute are defined in Section 4.9.7.3.

#### 4.9.7.3 Property Definitions
The TaskEditMode class MAY include additional property definitions which are implementation-defined.

#### 4.9.8 TaskEditModeEnum
The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses an editable mode of a task.
The TaskEditModeEnum class has attribute values:

- `localNamespace`
  - Value: icom_task

- `localName`
  - Value: TaskEditModeEnum

- `extendsFrom`
  - Value: icom_task:TaskEditMode

- `stereotype`
  - Value: primary

- `isEnumeration`
  - Value: TRUE

- `description`
  - Value: A mode that indicates whether a task is editable.

- `instances`
  - Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>

ICOM defines two task editable modes:

- `icom_task:OrganizerCopy`: a task is a copy created by an organizer who may update the properties such as start time, due time.

- `icom_task:AssigneeCopy`: a task is a copy delivered to an assignee who may only update the assignee properties such as completion time, participant status, percent completed.

### 4.10 Forum Module

#### 4.10.1 Discussion

**4.10.1.1 Description**

A discussion is an item in a discussion container.

**4.10.1.2 Class Definition**

The Discussion class is a mixin class that defines the characteristics of artifacts that can be elements of discussion containers.

The Discussion class has attribute values:

- `localNamespace`
  - Value: icom_forum
7360  
7361    **localName**
7362       Value: Discussion
7363
7364    **extendsFrom**
7365       Value: icom_core:Item
7366
7367    **stereotype**
7368       Value: mixin
7369
7370    **description**
7371       Value: Discussion is a mixin class that defines the characteristics of artifacts that can be placed in a discussion container.
7372
7374    **propertyDefinitions**
7375       The values for this attribute are defined in Section 4.10.1.3.

7376    **4.10.1.3 Property Definitions**
7377    The Discussion class inherits property definitions from super classes.
7378    The Discussion class MUST have the property definition:
7379
7380    **icom_forum:inReplyTo**
7381       Description: Another discussion object that a discussion object is replying to.
7382
7383       Required: False
7384       Inherited: False
7385       Property Type: icom_forum:Discussion
7386       Cardinality: Single
7387       Updatability: Read Write
7388
7389    The Discussion class MAY include additional property definitions which are implementation-defined.

7390
7391    **4.10.2 DiscussionContainer**

7392    **4.10.2.1 Description**
7393    A discussion container contains discussion items.

7394    **4.10.2.2 Class Definition**
7395    The DiscussionContainer class is a mixin class that defines the characteristics of folders that contain Discussion items.
The DiscussionContainer class has attribute values:

- **localNamespace**
  - Value: icom_forum

- **localName**
  - Value: DiscussionContainer

- **extendsFrom**
  - Value: icom_core:Container

- **stereotype**
  - Value: mixin

- **description**
  - Value: DiscussionContainer is a mixin class that defines the characteristics of folders that contain Discussion items.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.10.2.3.

### 4.10.2.3 Property Definitions

The DiscussionContainer class inherits property definitions from super classes.

The DiscussionContainer class MUST have the property definition:

- **icom_core:element**
  - Description: Elements of a discussion container.
  - Required: False
  - Inherited: True
  - Property Type: icom_forum:Discussion
  - Cardinality: Multi
  - Updatability: Read Only

The DiscussionContainer class MAY include additional property definitions which are implementation-defined.

### 4.10.3 DiscussionMessage

#### 4.10.3.1 Description

A discussion message is a message in a forum discussion thread.
4.10.3.2 Class Definition

The DiscussionMessage class has attribute values:

- **localNamespace**
  - Value: icom_forum

- **localName**
  - Value: DiscussionMessage

- **extendsFrom**
  - Value: icom_msg:Message, icom_forum:Discussion

- **stereotype**
  - Value: primary

- **description**
  - Value: Discussion message is a message in a forum discussion thread.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.10.3.3.

4.10.3.3 Property Definitions

The DiscussionMessage class inherits property definitions from super classes.

The DiscussionMessage class MUST have the property definition:

- **icom_forum:inReplyTo**
  - Description: Another discussion message that a discussion message is replying to.
  - Required: False
  - Inherited: True
  - Property Type: icom_forum:DiscussionMessage
  - Cardinality: Single
  - Updatability: Read Write

The DiscussionMessage class MAY include additional property definitions which are implementation-defined.

4.10.4 TopicContainer

4.10.4.1 Description

A topic container contains topics.
4.10.4.2 Class Definition

The TopicContainer class is a mixin class which defines the characteristics of folders that contain Topics.

The TopicContainer class has attribute values:

- **localNamespace**
  - Value: icom_forum

- **localName**
  - Value: TopicContainer

- **extendsFrom**
  - Value: icom_core:Container

- **stereotype**
  - Value: mixin

- **description**
  - Value: TopicContainer is a mixin class that defines the characteristics of folders that contain topics.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.10.4.3.

4.10.4.3 Property Definitions

The TopicContainer class inherits property definitions from super classes.

The TopicContainer class MUST have the property definitions:

- **icom_core:element**
  - Description: Elements of a topic container.
  - Required: False
  - Inherited: True
  - Property Type: icom_forum:Topic
  - Cardinality: Multi
  - Updatability: Read Only

The TopicContainer class MAY include additional property definitions which are implementation-defined.
4.10.5 Forum

4.10.5.1 Description
A forum contains sub-forums, topics, and announcements.

4.10.5.2 Class Definition
The Forum class has attribute values:

localNamespace
Value: icom_forum
localName
   Value: Forum

extendsFrom
   Value: icom_core:Folder, icom_forum:TopicContainer

 stereotype
   Value: primary

description
   Value: A forum contains sub-forums, topics, and announcements.

propertyDefinitions
   The values for this attribute are defined in Section 4.10.5.3.

4.10.5.3 Property Definitions
The Forum class inherits property definitions from super classes.
The Forum class MUST have the property definitions:

icom_forum:lastPost
   Description: The last posted discussion in a forum.
   Required: False
   Inherited: False
   Property Type: icom_forum:Discussion
   Cardinality: Single
   Updatability: Read Only

icom_forum:forum
   Description: Sub-forums of a forum.
   Required: False
   Inherited: False
   Property Type: icom_forum:Forum
   Cardinality: Multi
   Updatability: Read Only

icom_forum:topic
   Description: Topics of a forum.
   Required: False
   Inherited: False
   Property Type: icom_forum:Topic
   Cardinality: Multi
   Updatability: Read Only
4.10.6 Topic

4.10.6.1 Description
A topic contains conversations among forum participants. The discussions in a topic may be sorted in chronological order or threaded by reply.

4.10.6.2 Class Definition
The Topic class has attribute values:

- **localNamespace**
  - Value: icom_forum

- **localName**
  - Value: Topic

- **extendsFrom**
  - Value: icom_core:Folder, icom_forum:DiscussionContainer

- **stereotype**
  - Value: primary

- **description**
  - Value: A topic contains discussion threads.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.10.6.3.

4.10.6.3 Property Definitions
The Topic class inherits property definitions from super classes.

- **icom_core:element**
  - Description: Elements of a topic.
The Topic class MAY include additional property definitions which are implementation-defined.

4.10.7 Announcement

4.10.7.1 Description

An announcement contains time-sensitive discussion posts that are valid for a specified period of time, depending on activation and expiration times.

4.10.7.2 Class Definition

The Announcement class has attribute values:

- **localNamespace**
  - Value: icom_forum

- **localName**
  - Value: Announcement

- **extendsFrom**
  - Value: icom_forum:Topic
stereotype

Value: primary

description

Value: An announcement contains discussion items that are valid for a specified period of time.

propertyDefinitions

The values for this attribute are defined in Section 4.10.7.3.

4.10.7.3 Property Definitions

The Announcement class inherits property definitions from super classes.

The Announcement class MUST have the property definitions:

icom_forum:activationDate

Description: Date and time when an announcement becomes active.

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Write

icom_forum:expirationDate

Description: Date and time when an announcement expires.

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Write

icom_forum:announcementStatus

Description: Status of an announcement.

Required: True

Inherited: False

Property Type: icom_forum:AnnouncementStatus

Cardinality: Single

Updatability: Read Write

The Announcement class MAY include additional property definitions which are implementation-defined.
4.10.8 AnnouncementStatus

4.10.8.1 Description
An announcement status is status of an announcement.

4.10.8.2 Class Definition
The AnnouncementStatus class is a mixin class which defines status of an announcement.
The AnnouncementStatus class has attribute values:

```plaintext
localNamespace
  Value: icom_forum

localName
  Value: AnnouncementStatus

extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: AnnouncementStatus is a mixin class which defines status of an announcement.

propertyDefinitions
  The values for this attribute are defined in Section 4.10.8.3.
```

4.10.8.3 Property Definitions
The AnnouncementStatus class MAY include additional property definitions which are implementation-defined.

4.10.9 AnnouncementStatusEnum
The AnnouncementStatusEnum class is an enum class that enumerates the instances each of which expresses a status of announcement.
The AnnouncementStatusEnum class has attribute values:

```plaintext
localNamespace
  Value: icom_forum

localName
  Value: AnnouncementStatusEnum
```

4.10.9 AnnouncementStatusEnum
ICOM defines three announcement status:

- **icom_forum:Pending** an announcement is pending.
- **icom_forum:Active** an announcement is active.
- **icom_forum:Expired** an announcement is expired.

### 4.11 Conference Module

#### 4.11.1 Conference

#### 4.11.1.1 Description

A conference is a container that represents a durable context for conference sessions. It contains conference metadata, settings, and transcripts.

#### 4.11.1.2 Class Definition

The Conference class has attribute values:

- **localNamespace**
  - Value: icom_conf
- **localName**
  - Value: Conference
- **extendsFrom**
  - Value: icom_core:Folder
- **stereotype**
  - Value: primary
description
Value: A conference represents a durable context for online conference sessions.

propertyDefinitions
The values for this attribute are defined in Section 4.11.1.3.

4.11.1.3 Property Definitions
The Conference class inherits property definitions from super classes.
The Conference class MUST have the property definitions:

icom_core:organizer
Description: Organizer of a conference.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Single
Updatability: On Create

icom_conf:conferenceType
Description: Type of a conference.
Required: False
Inherited: False
Property Type: icom_conf:ConferenceType
Cardinality: Single
Updatability: Read Write

icom_conf:conferenceStatus
Description: Status of a conference.
Required: False
Inherited: False
Property Type: icom_conf:ConferenceStatus
Cardinality: Single
Updatability: Read Only

icom_conf:runningSession
Description: Current session of a conference.
Required: False
Inherited: False
Property Type: icom_conf:ConferenceSession
Cardinality: Single
Updatability: Read Only
icom_conf:conferenceSetting
Description: Configurable settings of a conference.
Required: False
Inherited: False
Property Type: icom_conf:ConferenceSetting
Cardinality: Single
Updatability: Read Only

icom_conf:transcript
Description: Transcripts from ended sessions of a conference.
Required: False
Inherited: False
Property Type: icom_doc:Document
Cardinality: Multi
Updatability: Read Write

icom_conf:scheduledStartDate
Description: Scheduled start date and time of a conference session.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Write

icom_conf:scheduledEndDate
Description: Scheduled end date and time of a conference session.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Write

The Conference class MAY include additional property definitions which are implementation-defined.
4.11.2 ConferenceType

4.11.2.1 Description

A conference type represents a category of conferences.

4.11.2.2 Class Definition

The ConferenceType class is a mixin class which defines a type of conference.

The ConferenceType class has attribute values:

- **localNamespace**
  - Value: `icom_conf`
7848     localName
7849          Value: ConferenceType
7850
7851     extendsFrom
7852          Value:
7853
7854     stereotype
7855          Value: mixin
7856
7857     description
7858          Value: ConferenceType is a mixin class which defines type of conference.
7859
7860     propertyDefinitions
7861          The values for this attribute are defined in Section 4.11.2.3.
7862
7863 **4.11.2.3 Property Definitions**
7864 The ConferenceType class MAY include additional property definitions which are implementation-defined.
7865
7866 **4.11.3 ConferenceTypeEnum**
7867 The ConferenceTypeEnum class is an enum class that enumerates the instances each of which expresses a type of a conference.
7868 The ConferenceTypeEnum class has attribute values:
7869
7870     localNamespace
7871          Value: icom_conf
7872
7873     localName
7874          Value: ConferenceTypeEnum
7875
7876     extendsFrom
7877          Value: icom_conf:ConferenceType
7878
7879     stereotype
7880          Value: primary
7881
7882     isEnumeration
7883          Value: TRUE
7884
7885     description
7886          Value: A type of a conference.
ICOM defines four conference types:

- `icom_conf:Impromptu` a conference session is started impromptu.
- `icom_conf:Scheduled` a conference session is scheduled.
- `icom_conf:ChatRoom` a conference is used for a chat room.
- `icom_conf:OtherConferenceType` a conference is of other type.

### 4.11.4 ConferenceStatus

#### 4.11.4.1 Description

A conference status is status of an online conference.

#### 4.11.4.2 Class Definition

The ConferenceStatus class is a mixin class which defines status of an online conference.

The ConferenceStatus class has attribute values:

```plaintext
localNamespace
  Value: icom_conf

localName
  Value: ConferenceStatus

extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: ConferenceStatus is a mixin class which defines status of an online conference.

propertyDefinitions
  The values for this attribute are defined in Section 4.11.4.3.
```

#### 4.11.4.3 Property Definitions

The ConferenceStatus class MAY include additional property definitions which are implementation-defined.
4.11.5 ConferenceStatusEnum

The ConferenceStatusEnum class is an enum class that enumerates the instances each of which expresses a status of a conference.

The ConferenceStateEnum class has attribute values:

```
localNamespace
    Value: icom_conf

localName
    Value: ConferenceStatusEnum

extendsFrom
    Value: icom_conf:ConferenceStatus

istereotype
    Value: primary

isEnumeration
    Value: TRUE

description
    Value: Status of a conference.

instances
    Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernating, icom_conf:Ended>
```

ICOM defines five conference status:

- **icom_conf:NotStarted** a conference session is not started.
- **icom_conf:WaitingForHost** a conference session is waiting for a host.
- **icom_conf:Running** a conference session is running.
- **icom_conf:Hibernating** a conference session is hibernating.
- **icom_conf:Ended** a conference session is ended.

4.11.6 ConferenceSession

4.11.6.1 Description

A conference session represents the metadata for a session of a conference.
4.11.6.2 Class Definition

The ConferenceSession class has attribute values:

- **localNamespace**
  - Value: icom_conf

- **localName**
  - Value: ConferenceSession

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: primary

- **description**
  - Value: A conference session represents the metadata for a session of a conference.

4.11.6.3 Property Definitions

The ConferenceSession class inherits property definitions from super classes. The ConferenceSession class MUST have the property definitions:

- **icom_core:description**
  - Description: Description of a conference session.
  - Required: False
  - Inherited: False
  - Property Type: String
  - Cardinality: Single
  - Updatability: Read Write

- **icom_core:startDate**
  - Description: Start date and time of a conference session.
  - Required: False
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
  - Updatability: Read Only
icom_core:endDate

Description: End date and time of a conference session.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Only

icom_conf:comment

Description: Comment on a conference session.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_conf:rating

Description: Rating of a conference session.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_conf:serverAddress

Description: Address of a server that hosts a conference session.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Only

icom_conf:endingReason

Description: Reason for ending a conference session.
Required: False
Inherited: False
Property Type: com:ConferenceSessionEndingReason
Cardinality: Single
Updatability: Read Only

The ConferenceSession class MAY include additional property definitions which are implementation-defined.
4.11.7 ConferenceSessionEndingReason

4.11.7.1 Description
A conference session ending reason is an indication of how a conference session ended.

4.11.7.2 Class Definition
The ConferenceSessionEndingReason class is a mixin class which defines an indication of how a conference session ended.

The ConferenceSessionEndingReason class has attribute values:

    localNamespace
    Value: icom_conf

    localName
    Value: ConferenceSessionEndingReason

    extendsFrom
    Value:

    stereotype
    Value: mixin

    description
    Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a conference session ended.

    propertyDefinitions
    The values for this attribute are defined in Section 4.11.7.3.

4.11.7.3 Property Definitions
The ConferenceSessionEndingReason class MAY include additional property definitions which are implementation-defined.

4.11.8 ConferenceSessionEndingReasonEnum
The ConferenceSessionEndingReasonEnum class is an enum class that enumerates the instances each of which expresses a reason for ending a conference session.

The ConferenceSessionEndingReasonEnum class has attribute values:

    localNamespace
    Value: icom_conf

    localName
    Value: ConferenceSessionEndingReasonEnum
ICOM defines four conference session states:

- **icom_conf:HostLeft** a conference session ended after the host left.
- **icom_conf:HostAborted** a conference session ended after the host aborted it.
- **icom_conf:NoHost** a conference session ended due to no one hosting.
- **icom_conf:Hibernating** a conference session is hibernating.

### 4.11.9 ConferenceSetting

#### 4.11.9.1 Description

A conference setting represents configuration settings for sessions of a conference.

#### 4.11.9.2 Class Definition

The ConferenceSetting class has attribute values:

- **localNamespace**
  - Value: icom_conf
- **localName**
  - Value: ConferenceSetting
- **extendsFrom**
  - Value:
- **stereotype**
  - Value: primary
description
   Value: A conference setting represents configuration settings for sessions of a conference.

propertyDefinitions
   The values for this attribute are defined in Section 4.11.9.3.

4.11.9.3 Property Definitions
   The ConferenceSetting class inherits property definitions from super classes.
   The ConferenceSetting class MUST have the property definitions:

   icom_meta:property
      Description: Configurable properties for a conference.
      Required: False
      Inherited: False
      Property Type: icom_meta:property
      Cardinality: Multi
      Updatability: Read Write

   icom_conf:participantRole
      Description: Role settings for conference participants.
      Required: False
      Inherited: False
      Property Type: icom_conf:ConferenceParticipantRole
      Cardinality: Multi
      Updatability: Read Write

The ConferenceSetting class MAY include additional property definitions which are implementation-defined.

4.11.10 ConferenceParticipantRole

4.11.10.1 Description
   A conference participant role defines roles settings for a conference participant.

4.11.10.2 Class Definition
   The ConferenceParticipantRole class has attribute values:

   localNamespace
      Value: icom_conf

   localName
      Value: ConferenceParticipantRole
extendsFrom
Value:

stereotype
Value: primary

description
Value: A conference participant role contains roles settings for a conference.

propertyDefinitions
The values for this attribute are defined in Section 4.11.10.3.

4.11.10.3 Property Definitions

The ConferenceParticipantRole class MUST have the property definitions:

icom_core:name
Description: Name of a role setting in a conference.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:participant
Description: One or more participants in a role setting.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
Updatability: Read Write

icom_meta:property
Description: Configurable properties for a role setting.
Required: False
Inherited: False
Property Type: icom_meta:Property
Cardinality: Multi
Updatability: Read Write

icom_conf:key
Description: One or more sign on keys to activate a role setting.
Required: False
Inherited: False
Property Type: String
Cardinality: Multi
Updatability: Read Write

**icom_conf:keyword**

Description: One or more key words to activate a role setting.
Required: False
Inherited: False
Property Type: String
Cardinality: Multi
Updatability: Read Write

The ConferenceParticipantRole class MAY include additional property definitions which are implementation-defined.
5 Conformance

5.1 Software Architecture or Framework Dependence

The ICOM specification does not presume a particular software architecture or framework for use of the ICOM model. Fulfillment of ICOM use case roles and accompanying responsibilities is implementation dependent.

5.2 Platform Provider Conformance

5.2.1 Platform Provider Conformance – No Extension Modules

An ICOM platform provider with no extension modules (Section 4):

a. SHALL conform to all mandatory statements and
b. MAY conform to optional statements

of the core ICOM model as defined in Section 3 of this standard.

5.2.2 Platform Provider Conformance – One or More Extension Modules

An ICOM platform provider with extension modules (Section 4):

a. SHALL conform to Section 5.2.1 and
b. SHALL conform to all mandatory statements and
c. MAY conform to optional statements

as defined in Section 4 for each extension module.

5.3 Service Provider Conformance

5.3.1 ICOM Service Provider – No Extension Modules

An ICOM service provider may provide one or more services defined in Section 3. For each such service provided, an ICOM service provider:

a. SHALL conform to all mandatory statements and
b. MAY conform to optional statements

for the classes, super classes, and related classes defined in Section 3 of this standard.

5.3.2 ICOM Service Provider – One or More Extension Modules

An ICOM service provider MAY support one or more extension modules as defined in Section 4 of this standard. For each service provided, an ICOM service provider:

a. SHALL conform to Section 5.3.1 (if an offered service is defined in Section 3) and
b. SHALL conform to all mandatory statements and
c. MAY conform to optional statements

as defined in Section 4 for that extension module.
5.4 ICOM Producer Conformance

5.4.1 ICOM Producer Conformance – No Extension Modules

An ICOM producer that produces no objects of a class conforming to Section 4:

a. SHALL conform to all mandatory statements and

b. MAY conform to optional statements

d. for the class and super classes thereof in Section 3 of this standard, for any object produced.

5.4.2 ICOM Producer Conformance – One or More Extension Modules

An ICOM producer that produces objects of a class conforming to Section 4:

a. SHALL conform to Section 5.4.1 and

b. SHALL conform to all mandatory statements and

c. MAY conform to optional statements

d. as defined in Section 4 for that extension module.

5.5 ICOM Consumer Conformance

5.5.1 ICOM Consumer Conformance – No Extension Modules

An ICOM consumer that consumes no objects of a class conforming to Section 4:

a. SHALL conform to all mandatory statements and

b. MAY conform to optional statements

d. for the class and super classes thereof in Section 3 of this standard, for any object consumed.

5.5.2 ICOM Consumer Conformance – Extension Modules

An ICOM consumer that consumes objects of a class conforming to Section 4:

a. SHALL conform to Section 5.5.1 and

b. SHALL conform to all mandatory statements and

c. MAY conform to optional statements

d. as defined in Section 4 for that extension module.
Appendix A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:
- Rafiul Ahad, Oracle Corporation
- Kenneth P. Baclawski, Northeastern University
- Eric S. Chan, Oracle Corporation
- Martin Chapman, Oracle Corporation
- Scott Conroy, Individual
- Stefan Decker, Digital Enterprise Research Institute (DERI)
- Laura Dragan, Digital Enterprise Research Institute (DERI)
- Patrick Durusau, Individual
- Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
- Deirdre Lee, Digital Enterprise Research Institute (DERI)
- Marc Pallot, ESoCE-NET
- Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
- Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
- Peter Saint-Andre, Cisco Systems, Inc.
- Ramesh Vasudevan, Oracle Corporation
- Peter Yim, Individual
# Appendix B. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSPRD 01</td>
<td>March 16, 2011</td>
<td>Eric S. Chan, Patrick Durusau</td>
<td>Committee Specification Draft for Public Review</td>
</tr>
<tr>
<td>CSPRD 02</td>
<td>November 8, 2011</td>
<td>Eric S. Chan, Patrick Durusau</td>
<td>Changes in response to public review comments.</td>
</tr>
<tr>
<td>CSPRD 03</td>
<td>March 20, 2012</td>
<td>Eric S. Chan, Patrick Durusau, Laura Dragan</td>
<td>Changes in response to TC members review comments.</td>
</tr>
<tr>
<td>CSPRD 04</td>
<td>June 26, 2012</td>
<td>Ken Baclawski</td>
<td>Add 4 additional attributes from grammar to PropertyDefinition metadata model, corrected spelling of Cardinality, renamed the address property of Addressable to entityAddress to avoid clashing with the address properties of EntityAddress and Participant, and specified the omitted namespaces of the superCategories of some of the enumerations.</td>
</tr>
<tr>
<td>CSPRD 05</td>
<td>October 15, 2012</td>
<td>Ken Baclawski, Eric S. Chan, Patrick Durusau</td>
<td>Change InstantMessage isAbstract to false, change PropertyType to optional in PropertyDefinition, change cardinality of superCategory property in Category to multi, add ClassDefinition, StereoType, StereoTypeEnum in icom_meta, add Figure 18 ClassDefinition UML diagram, remove EntityDefinition in icom_core. Updated the conformance clauses in Section 5.</td>
</tr>
</tbody>
</table>