



# IMS GLC Person Management Service Information Model

## Version 2.0

## Final Release Version 1.0

Date Issued: 30 June 2011

Latest version: <http://www.imsglobal.org/lis/>

### **IPR and Distribution Notices**

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS 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 IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: [http://www.imsglobal.org/ipr/imsipr\\_policyFinal.pdf](http://www.imsglobal.org/ipr/imsipr_policyFinal.pdf).

Copyright © 2011 IMS Global Learning Consortium. All Rights Reserved.

Use of this specification to develop products or services is governed by the license with IMS found on the IMS website: <http://www.imsglobal.org/license.html>.

Permission is granted to all parties to use excerpts from this document as needed in producing requests for proposals.

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

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

**© 2011 IMS Global Learning Consortium, Inc.  
All Rights Reserved.**

The IMS Logo is a trademark of the IMS Global Learning Consortium Inc.  
Documents Name: IMS Person Management Service Information Model – Revision: 30 June 2011

# Table of Contents

<b>LIST OF FIGURES.....</b>	<b>6</b>
<b>LIST OF TABLES.....</b>	<b>6</b>
<b>1 INTRODUCTION .....</b>	<b>10</b>
1.1 PERSON MANAGEMENT SERVICE OVERVIEW .....	10
1.2 SCOPE AND CONTEXT .....	10
1.3 STRUCTURE OF THIS DOCUMENT .....	11
1.4 VERSIONS 1 AND 2 COMPATIBILITY .....	11
1.5 NOMENCLATURE.....	12
1.6 REFERENCES .....	12
<b>2 PERSON MANAGEMENT SERVICES DESCRIPTION .....</b>	<b>14</b>
2.1 AN ABSTRACT REPRESENTATION .....	14
2.2 PERSON MANAGEMENT SERVICE ARCHITECTURE & SPECIFICATION MODEL .....	14
2.3 PERSON OBJECTS .....	15
2.4 SYNCHRONOUS & ASYNCHRONOUS SERVICES.....	15
2.5 HANDLING THE SERVICE STATUS CODES.....	16
<b>3 BEHAVIORAL MODEL .....</b>	<b>18</b>
3.1 SERVICE DEFINITION .....	18
3.2 PERSONMANAGER INTERFACE DESCRIPTION.....	18
3.2.1 <i>CreatePerson()</i> Operation.....	20
3.2.2 <i>CreateByProxyPerson()</i> Operation .....	22
3.2.3 <i>DeletePerson()</i> Operation .....	24
3.2.4 <i>ReadPerson()</i> Operation .....	25
3.2.5 <i>ReadCorePerson()</i> Operation.....	26
3.2.6 <i>ReadAllPersonIds()</i> Operation.....	27
3.2.7 <i>ReadPersonIdsFromSavePoint()</i> Operation.....	28
3.2.8 <i>ReadPersons()</i> Operation.....	29
3.2.9 <i>ReadPersonsFromSavePoint()</i> Operation.....	30
3.2.10 <i>UpdatePerson()</i> Operation.....	31
3.2.11 <i>ReplacePerson()</i> Operation.....	33
3.2.12 <i>DiscoverPersonIds()</i> Operation .....	35
3.2.13 <i>ChangePersonIdentifier()</i> Operation .....	36
3.3 PERSON OBJECT STATE MACHINE .....	37
<b>4 INTERFACE DATA MODEL.....</b>	<b>39</b>
4.1 GUID CLASS DESCRIPTION .....	39
4.2 GUIDSET CLASS DESCRIPTION .....	39

4.3	PERSONRECORD CLASS DESCRIPTION .....	39
4.4	PERSONRECORDSET CLASS DESCRIPTION .....	39
4.5	PERSONCORE CLASS DESCRIPTION .....	39
4.5.1	<i>SourcedId Attribute Description</i> .....	40
4.5.2	<i>Formname Attribute Description</i> .....	40
4.5.3	<i>UserId Attribute Description</i> .....	40
4.6	QUERYOBJECT CLASS DESCRIPTION .....	41
4.7	SEQUENCEIDENTIFIER CLASS DESCRIPTION .....	41
4.8	STATUSINFO CLASS DESCRIPTION .....	41
<b>5</b>	<b>END SYSTEM DATA MODEL .....</b>	<b>42</b>
5.1	KEY TERMS AND CONCEPTS .....	42
5.2	PERSONDATABASE CLASS DESCRIPTION .....	45
5.2.1	<i>PersonRecord Attribute Description</i> .....	46
5.3	PERSONRECORD CLASS DESCRIPTION .....	46
5.3.1	<i>SourcedGUID Attribute Description</i> .....	47
5.4	SOURCEDGUID CLASS DESCRIPTION .....	47
5.4.1	<i>RefAgentInstanceID Attribute Description</i> .....	47
5.4.2	<i>SourcedId Attribute Description</i> .....	48
5.5	PERSON CLASS DESCRIPTION .....	48
5.5.1	<i>Formname Attribute Description</i> .....	49
5.5.2	<i>Name Attribute Description</i> .....	49
5.5.3	<i>Address Attribute Description</i> .....	50
5.5.4	<i>ContactInfo Attribute Description</i> .....	50
5.5.5	<i>Demographics Attribute Description</i> .....	50
5.5.6	<i>Agent Attribute Description</i> .....	51
5.5.7	<i>Roles Attribute Description</i> .....	51
5.5.8	<i>DataSource Attribute Description</i> .....	51
5.5.9	<i>Extension Attribute Description</i> .....	52
5.6	FORMNAME CLASS DESCRIPTION .....	53
5.6.1	<i>FormnameType Attribute Description</i> .....	54
5.6.2	<i>FormattedName Attribute Description</i> .....	54
5.7	NAME CLASS DESCRIPTION .....	55
5.7.1	<i>NameType Attribute Description</i> .....	56
5.7.2	<i>PartName Attribute Description</i> .....	56
5.8	ADDRESS CLASS DESCRIPTION .....	57
5.8.1	<i>AddressType Attribute Description</i> .....	58
5.8.2	<i>AddressPart Attribute Description</i> .....	58
5.9	CONTACTINFO CLASS DESCRIPTION .....	59
5.10	DEMOGRAPHICS CLASS DESCRIPTION .....	61
5.10.1	<i>DemographicsType Attribute Description</i> .....	62
5.10.2	<i>Dates Attribute Description</i> .....	63

5.10.3	<i>Gender Attribute Description</i> .....	63
5.10.4	<i>DemographicInfo Attribute Description</i> .....	64
5.11	REPRESENTATION CLASS DESCRIPTION .....	64
5.11.1	<i>RepresentationType Attribute Description</i> .....	65
5.12	AGENT CLASS DESCRIPTION .....	66
5.12.1	<i>AgentType Attribute Description</i> .....	67
5.12.2	<i>AgentId Attribute Description</i> .....	67
5.12.3	<i>Agentdomain Attribute Description</i> .....	68
5.12.4	<i>Description Attribute Description</i> .....	68
5.13	ENTERPRISE ROLES CLASS DESCRIPTION .....	69
5.13.1	<i>EnterpriseRolesType Attribute Description</i> .....	70
5.13.2	<i>SystemRole Attribute Description</i> .....	70
5.13.3	<i>InstitutionRole Attribute Description</i> .....	71
5.13.4	<i>Enrollment Attribute Description</i> .....	71
5.13.5	<i>UserId Attribute Description</i> .....	72
5.14	USERID CLASS DESCRIPTION .....	72
5.14.1	<i>UserIdValue Attribute Description</i> .....	72
5.14.2	<i>UserIdType Attribute Description</i> .....	73
5.14.3	<i>Password Attribute Description</i> .....	73
5.14.4	<i>PwEncryption Attribute Description</i> .....	73
5.14.5	<i>AuthenticationType Attribute Description</i> .....	74
5.15	INSTITUTIONROLE CLASS DESCRIPTION .....	74
5.15.1	<i>Institutionrolevalue Attribute Description</i> .....	75
5.15.2	<i>Primaryroletype Attribute Description</i> .....	75
5.16	COMMON CLASSES DESCRIPTIONS .....	76
5.16.1	<i>BaseValueToken Class Description</i> .....	77
5.16.2	<i>BaseValueSingle Class Description</i> .....	78
5.16.3	<i>Description Class Description</i> .....	80
5.16.4	<i>FullDescription Class Description</i> .....	81
5.16.5	<i>Text Class Description</i> .....	83
5.16.6	<i>IMSExtension Class Description</i> .....	84
5.16.7	<i>ExtensionField Class Description</i> .....	85
<b>6</b>	<b>EXTENDING AND PROFILING THE SERVICE</b> .....	<b>87</b>
6.1	PROPRIETARY EXTENSIONS .....	87
6.1.1	<i>Proprietary Operations</i> .....	87
6.1.2	<i>Proprietary Data Elements</i> .....	87
6.2	PROFILING THE SERVICE .....	87
	<b>APPENDIX A – SERVICE STATUS CODES</b> .....	<b>88</b>
	<b>APPENDIX B – SET OF VOCABULARIES</b> .....	<b>90</b>
B1	SET OF DEFINED VOCABULARIES .....	90
B1.1	<i>FormnameType Vocabulary</i> .....	90

<i>B1.2 NameType Vocabulary .....</i>	<i>90</i>
<i>B1.3 PartName Vocabulary.....</i>	<i>91</i>
<i>B1.4 AddressType Vocabulary.....</i>	<i>92</i>
<i>B1.5 AddressPart Vocabulary .....</i>	<i>93</i>
<i>B1.6 ContactinfoType Vocabulary.....</i>	<i>94</i>
<i>B1.7 DemographicsType Vocabulary .....</i>	<i>96</i>
<i>B1.8 DemographicInfo Vocabulary.....</i>	<i>97</i>
<i>B1.9 EventDate Vocabulary.....</i>	<i>97</i>
<i>B1.10 RepresentationType Vocabulary.....</i>	<i>98</i>
<i>B1.11 AgentType Vocabulary .....</i>	<i>99</i>
<i>B1.12 EnterpriserolesType Vocabulary.....</i>	<i>99</i>
<i>B1.13 SystemRole Vocabulary.....</i>	<i>100</i>
<i>B1.14 Institutionrolevalue Vocabulary.....</i>	<i>100</i>
<i>B1.15 Enrollment Vocabulary .....</i>	<i>101</i>
<i>B1.16 FieldType Vocabulary.....</i>	<i>102</i>
<i>B1.17 Language Vocabulary .....</i>	<i>102</i>
<b>B2 USING VOCABULARIES FOR THE METADATA CLASS .....</b>	<b>103</b>
<b>B3 USING VOCABULARIES FOR THE IMSEXTENSION CLASS .....</b>	<b>103</b>
<b>APPENDIX C – FILE-BASED DATA EXCHANGE.....</b>	<b>104</b>
<b>ABOUT THIS DOCUMENT.....</b>	<b>105</b>
LIST OF CONTRIBUTORS.....	105
<b>REVISION HISTORY .....</b>	<b>106</b>
<b>INDEX .....</b>	<b>107</b>

## List of Figures

FIGURE 2.1 PERSON MANAGEMENT SERVICE ARCHITECTURE MODEL. ....	14
FIGURE 2.2 SYNCHRONOUS SERVICE. ....	15
FIGURE 2.3 ASYNCHRONOUS SERVICE. ....	16
FIGURE 3.1 PERSONMANAGEMENTSERVICE INTERFACE DEFINITION. ....	18
FIGURE 3.2 STATE MACHINE FOR A ‘PERSON’ OBJECT. ....	37
FIGURE 4.1 THE PERSONCORE CLASS DIAGRAM. ....	39
FIGURE 5.1 PERSONDATABASE CLASS DIAGRAM. ....	45
FIGURE 5.2 PERSON CLASS DIAGRAM. ....	48
FIGURE 5.3 FORMNAME CLASS DIAGRAM. ....	53
FIGURE 5.4 NAME CLASS DIAGRAM. ....	55
FIGURE 5.5 ADDRESS CLASS DIAGRAM. ....	57
FIGURE 5.6 CONTACTINFO CLASS DIAGRAM. ....	59
FIGURE 5.7 DEMOGRAPHICS CLASS DIAGRAM. ....	61
FIGURE 5.8 AGENT CLASS DIAGRAM. ....	66
FIGURE 5.9 ENTERPRISEROLES CLASS DIAGRAM. ....	69
FIGURE 5.10 COMMON CLASSES DIAGRAM. ....	76
FIGURE C.1 PERSONRECORD CLASS DIAGRAM FOR FILE-BASED DATA EXCHANGE. ....	104

## List of Tables

TABLE 3.1 SUMMARY OF OPERATIONS FOR PERSONMANAGER. ....	19
TABLE 3.2 STATUS CODES FOR THE ‘CREATEPERSON’ OPERATION. ....	20
TABLE 3.3 STATUS CODES FOR THE ‘CREATEBYPROXYPERSON’ OPERATION. ....	22
TABLE 3.4 STATUS CODES FOR THE ‘DELETEPERSON’ OPERATION. ....	24
TABLE 3.5 STATUS CODES FOR THE ‘READPERSON’ OPERATION. ....	25
TABLE 3.6 STATUS CODES FOR THE ‘READCOREPERSON’ OPERATION. ....	26
TABLE 3.7 STATUS CODES FOR THE ‘READALLPERSONIDS’ OPERATION. ....	27
TABLE 3.8 STATUS CODES FOR THE ‘READPERSONIDSFROMSAVEPOINT’ OPERATION. ....	28
TABLE 3.9 STATUS CODES FOR THE ‘READPERSONS’ OPERATION. ....	29
TABLE 3.10 STATUS CODES FOR THE ‘READPERSONSFROMSAVEPOINT’ OPERATION. ....	30
TABLE 3.11 STATUS CODES FOR THE ‘UPDATEPERSON’ OPERATION. ....	31
TABLE 3.12 STATUS CODES FOR THE ‘REPLACEPERSON’ OPERATION. ....	33
TABLE 3.13 STATUS CODES FOR THE ‘DISCOVERPERSONIDS’ OPERATION. ....	35
TABLE 3.14 STATUS CODES FOR THE ‘CHANGEPERSONIDENTIFIER’ OPERATION. ....	36
TABLE 4.1 DESCRIPTION OF THE ‘SOURCEDId’ ATTRIBUTE FOR THE PERSONCORE CLASS. ....	40
TABLE 4.2 DESCRIPTION OF THE ‘FORMNAME’ ATTRIBUTE FOR THE PERSONCORE CLASS. ....	40
TABLE 4.3 DESCRIPTION OF THE ‘USERId’ ATTRIBUTE FOR THE PERSONCORE CLASS. ....	40
TABLE 5.1 CLASS DESCRIPTORS ....	42

TABLE 5.2 DESCRIPTION OF THE ‘PERSONDATABASE’ CLASS. ....	45
TABLE 5.3 DESCRIPTION OF THE ‘PERSONRECORD’ ATTRIBUTE FOR THE PERSONDATABASE CLASS.....	46
TABLE 5.4 DESCRIPTION OF THE ‘PERSONRECORD’ CLASS.....	46
TABLE 5.5 DESCRIPTION OF THE ‘SOURCEDGUID’ ATTRIBUTE FOR THE PERSONRECORD CLASS.....	47
TABLE 5.6 DESCRIPTION OF THE SOURCEDGUID CLASS. ....	47
TABLE 5.7 DESCRIPTION OF THE ‘REFAGENTINSTANCEID’ ATTRIBUTE FOR THE SOURCEDGUID CLASS. ....	47
TABLE 5.8 DESCRIPTION OF THE ‘SOURCEDId’ ATTRIBUTE FOR THE SOURCEDGUID CLASS.....	48
TABLE 5.9 DESCRIPTION OF THE ‘PERSON’ CLASS. ....	49
TABLE 5.10 DESCRIPTION OF THE ‘FORMNAME’ ATTRIBUTE FOR THE PERSON CLASS.....	49
TABLE 5.11 DESCRIPTION OF THE ‘NAME’ ATTRIBUTE FOR THE PERSON CLASS. ....	49
TABLE 5.12 DESCRIPTION OF THE ‘ADDRESS’ ATTRIBUTE FOR THE PERSON CLASS. ....	50
TABLE 5.13 DESCRIPTION OF THE ‘CONTACTINFO’ ATTRIBUTE FOR THE PERSON CLASS. ....	50
TABLE 5.14 DESCRIPTION OF THE ‘DEMOGRAPHICS’ ATTRIBUTE FOR THE PERSON CLASS.....	50
TABLE 5.15 DESCRIPTION OF THE ‘AGENT’ ATTRIBUTE FOR THE PERSON CLASS. ....	51
TABLE 5.16 DESCRIPTION OF THE ‘ROLES’ ATTRIBUTE FOR THE PERSON CLASS.....	51
TABLE 5.17 DESCRIPTION OF THE ‘DATASOURCE’ ATTRIBUTE FOR THE PERSON CLASS. ....	51
TABLE 5.18 DESCRIPTION OF THE ‘EXTENSION’ ATTRIBUTE FOR THE PERSON CLASS. ....	52
TABLE 5.19 DESCRIPTION OF THE ‘FORMNAME’ CLASS. ....	53
TABLE 5.20 DESCRIPTION OF THE ‘FORMNAMEType’ ATTRIBUTE FOR THE FORMNAME CLASS. ....	54
TABLE 5.21 DESCRIPTION OF THE ‘FORMATTEDNAME’ ATTRIBUTE FOR THE FORMNAME CLASS.....	54
TABLE 5.22 DESCRIPTION OF THE ‘NAME’ CLASS.....	55
TABLE 5.23 DESCRIPTION OF THE ‘NAMEType’ ATTRIBUTE FOR THE NAME CLASS.....	56
TABLE 5.24 DESCRIPTION OF THE ‘PARTNAME’ ATTRIBUTE FOR THE NAME CLASS. ....	56
TABLE 5.25 DESCRIPTION OF THE ‘ADDRESS’ CLASS.....	57
TABLE 5.26 DESCRIPTION OF THE ‘ADDRESSType’ ATTRIBUTE FOR THE ADDRESS CLASS. ....	58
TABLE 5.27 DESCRIPTION OF THE ‘ADDRESSPART’ ATTRIBUTE FOR THE ADDRESS CLASS. ....	58
TABLE 5.28 DESCRIPTION OF THE ‘CONTACTINFO’ CLASS. ....	59
TABLE 5.29 DESCRIPTION OF THE ‘CONTACTINFOType’ ATTRIBUTE FOR THE CONTACTINFO CLASS. ....	60
TABLE 5.30 DESCRIPTION OF THE ‘CONTACTINFOValue’ ATTRIBUTE FOR THE CONTACTINFO CLASS.....	60
TABLE 5.31 DESCRIPTION OF THE ‘DEMOGRAPHICS’ CLASS. ....	61
TABLE 5.32 DESCRIPTION OF THE ‘DEMOGRAPHICSType’ ATTRIBUTE FOR THE DEMOGRAPHICS CLASS. ....	62
TABLE 5.33 DESCRIPTION OF THE ‘REPRESENTATION’ ATTRIBUTE FOR THE DEMOGRAPHICS CLASS.....	62
TABLE 5.34 DESCRIPTION OF THE ‘EVENTDATE’ ATTRIBUTE FOR THE DEMOGRAPHICS CLASS. ....	63
TABLE 5.35 DESCRIPTION OF THE ‘GENDER’ ATTRIBUTE FOR THE DEMOGRAPHICS CLASS. ....	63
TABLE 5.36 DESCRIPTION OF THE ‘DEMOGRAPHICINFO’ ATTRIBUTE FOR THE DEMOGRAPHICS CLASS. ....	64
TABLE 5.37 DESCRIPTION OF THE ‘REPRESENTATION’ CLASS. ....	64
TABLE 5.38 DESCRIPTION OF THE ‘REPRESENTATIONType’ ATTRIBUTE OF THE REPRESENTATION CLASS. ....	65
TABLE 5.39 DESCRIPTION OF THE ‘DATE’ ATTRIBUTE OF THE REPRESENTATION CLASS.....	65
TABLE 5.40 DESCRIPTION OF THE ‘DESCRIPTION’ ATTRIBUTE OF THE REPRESENTATION CLASS. ....	65
TABLE 5.41 DESCRIPTION OF THE ‘AGENT’ CLASS. ....	66
TABLE 5.42 DESCRIPTION OF THE ‘AGENTType’ ATTRIBUTE FOR THE AGENT CLASS. ....	67
TABLE 5.43 DESCRIPTION OF THE ‘AGENTId’ ATTRIBUTE FOR THE AGENT CLASS.....	67

TABLE 5.44 DESCRIPTION OF THE ‘AGENTDOMAIN’ ATTRIBUTE FOR THE AGENT CLASS.....	68
TABLE 5.45 DESCRIPTION OF THE ‘DESCRIPTION’ ATTRIBUTE FOR THE AGENT CLASS.....	68
TABLE 5.46 DESCRIPTION OF THE ‘ENTERPRISEROLES’ CLASS.....	69
TABLE 5.47 DESCRIPTION OF THE ‘ENTERPRISEROLES TYPE’ ATTRIBUTE FOR THE ENTERPRISEROLES CLASS.....	70
TABLE 5.48 DESCRIPTION OF THE ‘SYSTEMROLE’ ATTRIBUTE FOR THE ENTERPRISEROLES CLASS.....	70
TABLE 5.49 DESCRIPTION OF THE ‘INSTITUTIONROLE’ ATTRIBUTE FOR THE ENTERPRISEROLES CLASS.....	71
TABLE 5.50 DESCRIPTION OF THE ‘ENROLLMENT’ ATTRIBUTE FOR THE ENTERPRISEROLES CLASS.....	71
TABLE 5.51 DESCRIPTION OF THE ‘USERID’ ATTRIBUTE FOR THE ENTERPRISEROLES CLASS.....	72
TABLE 5.52 DESCRIPTION OF THE ‘USERID’ CLASS.....	72
TABLE 5.53 DESCRIPTION OF THE ‘USERIDVALUE’ ATTRIBUTE FOR THE USERID CLASS.....	72
TABLE 5.54 DESCRIPTION OF THE ‘USERIDTYPE’ ATTRIBUTE FOR THE USERID CLASS.....	73
TABLE 5.55 DESCRIPTION OF THE ‘PASSWORD’ ATTRIBUTE FOR THE USERID CLASS.....	73
TABLE 5.56 DESCRIPTION OF THE ‘PWENCRYPTION’ ATTRIBUTE FOR THE USERID CLASS.....	73
TABLE 5.57 DESCRIPTION OF THE ‘AUTHENTICATIONTYPE’ ATTRIBUTE FOR THE USERID CLASS.....	74
TABLE 5.58 DESCRIPTION OF THE ‘INSTITUTIONROLE’ CLASS.....	74
TABLE 5.59 DESCRIPTION OF THE ‘INSTITUTIONROLEVALUE’ ATTRIBUTE FOR THE INSTITUTIONROLE CLASS.....	75
TABLE 5.60 DESCRIPTION OF THE ‘PRIMARYROLETYPE’ ATTRIBUTE FOR THE INSTITUTIONROLE CLASS.....	75
TABLE 5.61 DESCRIPTION OF THE ‘BASEVALUETOKEN’ CLASS.....	77
TABLE 5.62 DESCRIPTION OF THE ‘INSTANCEIDENTIFIER’ ATTRIBUTE FOR THE BASEVALUETOKEN CLASS.....	77
TABLE 5.63 DESCRIPTION OF THE ‘INSTANCEVOCABULARY’ ATTRIBUTE FOR THE BASEVALUETOKEN CLASS.....	77
TABLE 5.64 DESCRIPTION OF THE ‘INSTANCEVALUE’ ATTRIBUTE FOR THE BASEVALUETOKEN CLASS.....	78
TABLE 5.65 DESCRIPTION OF THE ‘BASEVALUESINGLE’ CLASS.....	78
TABLE 5.66 DESCRIPTION OF THE ‘INSTANCEIDENTIFIER’ ATTRIBUTE FOR THE BASEVALUESINGLE CLASS.....	78
TABLE 5.67 DESCRIPTION OF THE ‘INSTANCEVOCABULARY’ ATTRIBUTE FOR THE BASEVALUESINGLE CLASS.....	79
TABLE 5.68 DESCRIPTION OF THE ‘INSTANCENAME’ ATTRIBUTE FOR THE BASEVALUESINGLE CLASS.....	79
TABLE 5.69 DESCRIPTION OF THE ‘INSTANCEVALUE’ ATTRIBUTE FOR THE BASEVALUESINGLE CLASS.....	79
TABLE 5.70 DESCRIPTION OF THE ‘DESCRIPTION’ CLASS.....	80
TABLE 5.71 DESCRIPTION OF THE ‘SHORTDESCRIPTION’ ATTRIBUTE FOR THE DESCRIPTION CLASS.....	80
TABLE 5.72 DESCRIPTION OF THE ‘LONGDESCRIPTION’ ATTRIBUTE FOR THE DESCRIPTION CLASS.....	80
TABLE 5.73 DESCRIPTION OF THE ‘FULLDESCRIPTION’ ATTRIBUTE FOR THE DESCRIPTION CLASS.....	81
TABLE 5.74 DESCRIPTION OF THE ‘FULLDESCRIPTION’ CLASS.....	81
TABLE 5.75 DESCRIPTION OF THE ‘MEDIAMODE’ ATTRIBUTE FOR THE FULLDESCRIPTION CLASS.....	81
TABLE 5.76 DESCRIPTION OF THE ‘CONTENTREFTYPE’ ATTRIBUTE FOR THE FULLDESCRIPTION CLASS.....	82
TABLE 5.77 DESCRIPTION OF THE ‘MIMETYPE’ ATTRIBUTE FOR THE FULLDESCRIPTION CLASS.....	82
TABLE 5.78 DESCRIPTION OF THE ‘DESCRIPTIONTEXT’ ATTRIBUTE FOR THE FULLDESCRIPTION CLASS.....	82
TABLE 5.79 DESCRIPTION OF THE ‘TEXT’ CLASS.....	83
TABLE 5.80 DESCRIPTION OF THE ‘LANGUAGE’ ATTRIBUTE FOR THE TEXT CLASS.....	83
TABLE 5.81 DESCRIPTION OF THE ‘TEXTSTRING’ ATTRIBUTE FOR THE TEXT CLASS.....	83
TABLE 5.82 DESCRIPTION OF THE IMSEXTENSION CLASS.....	84
TABLE 5.83 DESCRIPTION OF THE ‘EXTENSIONNAMEVOCABULARY’ ATTRIBUTE FOR THE IMSEXTENSION CLASS.....	84
TABLE 5.84 DESCRIPTION OF THE ‘EXTENSIONTYPEVOCABULARY’ ATTRIBUTE FOR THE IMSEXTENSION CLASS.....	84
TABLE 5.85 DESCRIPTION OF THE ‘EXTENSIONFIELD’ ATTRIBUTE FOR THE IMSEXTENSION CLASS.....	85



TABLE 5.86 DESCRIPTION OF THE EXTENSIONFIELD CLASS.....	85
TABLE 5.87 DESCRIPTION OF THE 'FIELDNAME' ATTRIBUTE FOR THE EXTENSIONFIELD CLASS.....	85
TABLE 5.88 DESCRIPTION OF THE 'FIELDTYPE' ATTRIBUTE FOR THE EXTENSIONFIELD CLASS. ....	86
TABLE 5.89 DESCRIPTION OF THE 'FIELDVALUE' ATTRIBUTE FOR THE EXTENSIONFIELD CLASS. ....	86
TABLE A.1 STATUS CODES FOR THE PERSONMANAGER INTERFACE OPERATIONS. ....	88
TABLE A.2 COMMON STATUS CODES FOR THE SERVICE OPERATIONS. ....	89
TABLE B1.1 VOCABULARY FOR 'FORMNAMEType'. ....	90
TABLE B1.2 VOCABULARY FOR 'NAMEType'. ....	90
TABLE B1.3 VOCABULARY FOR TYPE OF 'PARTNAME'. ....	91
TABLE B1.4 VOCABULARY FOR THE 'ADDRESSType'. ....	92
TABLE B1.5 VOCABULARY FOR THE TYPE OF 'ADDRESSPART'. ....	93
TABLE B1.6 VOCABULARY FOR THE 'CONTACTINFOType'. ....	94
TABLE B1.7 VOCABULARY FOR THE 'DEMOGRAPHICSType'. ....	96
TABLE B1.8 VOCABULARY FOR THE 'DEMOGRAPHICINFO'. ....	97
TABLE B1.9 VOCABULARY FOR THE 'EVENTDATE'. ....	97
TABLE B1.10 VOCABULARY FOR THE 'REPRESENTATIONType'. ....	98
TABLE B1.11 VOCABULARY FOR THE 'AGENTType'. ....	99
TABLE B1.12 VOCABULARY FOR THE 'ENTERPRISEROLESType'. ....	99
TABLE B1.13 VOCABULARY FOR THE 'SYSTEMROLE'. ....	100
TABLE B1.14 VOCABULARY FOR THE 'INSTITUTIONROLEVALUE'. ....	100
TABLE B1.15 VOCABULARY FOR THE 'SYSTEMROLE'. ....	101
TABLE B1.16 VOCABULARY FOR THE 'FIELDType'. ....	102

# 1 Introduction

## 1.1 Person Management Service Overview

The Person Management Service (PMS) specification is the definition of how systems manage the exchange of information that describes people. The Person Management Service specification is constructed following the recommendations documented in the IMS Global Learning Consortium (GLC) Abstract Framework (IAF) [IAF, 03a], [IAF, 03b], [IAF, 03c]. This means that this specification is based upon the concepts of:

- Interoperability – Person Management Service focuses on the exchange of Person(s) information between systems. There are no definitions in the specification on how the data is managed within the systems;
- Service-oriented – Person Management Service defines the exchange of information in terms of the services being supplied by the collaboration of the systems;
- Component-based – for example, the Person Management Service is combined with the Group Management Service, Membership Management Service, Course Management Service, Outcomes Management Service and Bulk Data Exchange Management Service to provide the Learning Information Services [LIS, 011a];
- Layering – the Person Management Service is a part of the Application Services layer but it interacts with the services available in the Common Services layer e.g., authentication;
- Behaviors and Data Models – the Person Management Service are defined in terms of their behaviors and data models. The behaviors cause changes in the state of the data model and the state of the data model will only be altered as a result of a clearly defined behavior;
- Multiple Bindings – the Person Management Service information model is defined using the Unified Modeling Language (UML). This enables reliable mapping of the information model into a range of different bindings. The binding of immediate importance is to the Web Services Description Language (WSDL);
- Adoption – whenever appropriate, the Person Management Service specification makes use of other IMS GLC and non-IMS GLC standards and specifications.

The aim of this specification is to enable systems to exchange information about a person. The defined service can be included in any type of system e.g., Student Information System, Human Resources System, etc. A ‘person’ object is composed of six categories of information: name, address, contact information, demographics, agent and enterprise roles. The service operations provide the capability for creating, deleting, reading, writing and simple searching of person objects.

## 1.2 Scope and Context

This document is the IMS GLC Person Management Services Information Model v2.0 and as such it is used as the basis for the development of the following documents:

- a) IMS GLC Person Management Service WSDL Binding v2.0 [PMS, 11] – the description of the WSDL binding of the Information Model;

The core use-cases for the Person Management Service are described in the Learning Information Services Specification [LIS, 11b]. This Person Management Service specification supersedes v1.0. The ‘identification’ object within the IMS GLC Learner Information Package (LIP) v1.0 specification [LIP, 01] has been integrated with the ‘person’ object from the IMS GLC Person Management Service v1.0 specification.

This information model defines the Person Management Service Abstract Application Programming Interface (a-API). The Learning Information Services specification, of which the Person Management Service is a component, is a series of behavioral models that define how the data models are to be manipulated. These behavioral models are described using the Unified Modeling Language (UML) [SDN07, 07].

## 1.3 Structure of this Document

The structure of this document is:

2. PERSON MANAGEMENT SERVICE DESCRIPTION	The description of the overall structure and operation of the Person Management Service. This includes the description of the architectural model and the domain object model;
3. BEHAVIORAL MODEL	The definition of the operations of a Person Management Service application service. This focuses on the description of the behaviors supported by the service;
4. INTERFACE DATA MODEL	The definition of the data models exchanged between the Person Management Service End Systems. These are the parameters exchanged across the interoperability interface;
5. END SYSTEM DATA MODEL	The definition of the data models for the Person Management Service End Systems. This addresses the persistence of the data with respect to interoperability;
6. EXTENDING AND PROFILING THE SERVICE	Identification of the ways in which the Person Management Service can be extended both in terms of the addition of new constituent services and proprietary extensions to a service;
APPENDIX A SERVICE STATUS CODES	A summary list of the status codes, and their causes, that can be returned by each of the operations forming the Person Management Service.
APPENDIX B SET OF VOCABULARIES	The set of vocabularies that have been defined for this service;
APPENDIX C FILE-BASED DATA EXCHANGE	The out-of-band file exchange used in response to receiving a URL for an external data file that contains the request data.

## 1.4 Versions 1 and 2 Compatibility

The functional changes in version 2 compared to version 1 are:

- a) A single service interface is used. With the exception of the ‘ReadPersons’ operation all of the operations in the original ‘PersonsManagement’ interface have been removed;
- b) The ‘ReadPersons’ operation has been changed such that it returns a single ‘statusInfo’ object;
- c) New service operations have been added, namely:-
  - ReadCorePerson – to read information that is defined as the ‘core’ Person
  - ReadAllPersonIds – to read all of the sourcedIds allocated in the target system
  - ReadPersonIdsFromSavePoint – to read all of the sourcedIds for ‘person’ objects that have been altered since the defined reference point
  - ReadPersonsFromSavePoint – to read all of the ‘person’ objects, that have been altered since the defined reference point
  - DiscoverPersonIds – to provide the sourcedIds of the ‘person’ objects that are identified by the completion of the requested query operation;
- d) Use of a file transfer protocol to obtain the data contained in an external data file is introduced;
- e) The core data model has been extended to support both the PMSv1.0 and the IMS GLC LIPv1.0 ‘identification’ data models. The data model has also been modified to use external vocabularies.

The release of the Person Management Services 2.0 creates the issue of compatibility between version 1 and version 2 implementations. Compatibility issues occur when:

- a) A version 1 PMS implementation initiates data exchange with a version 2 implementation;
- b) A version 2 PMS implementation initiates data exchange with a version 1 implementation.

The binding of the Information Model recommends that the Uniform Resource Locator (URL) for the messaging actions is dependent on the type and version number of the source specification: in such a case it is not possible for cross-interaction between implementations of version 1 and 2. However, if a common URL is used then cross-interaction becomes possible. The definition of the behavior for interactions between different versions is beyond the scope of this specification.

## 1.5 Nomenclature

API	Application Programming Interface
a-API	Abstract Application Programming Interface
CM/DN	Contributing Member/Developer Network
FTP	File Transfer Protocol
IAF	IMS GLC Abstract Framework
IMS GLC	IMS Global Learning Consortium Inc.
LIP	Learner Information Package
LIS	Learning Information Services
PIM	Platform Independent Model
PMS	Person Management Service
PSM	Platform Specific Model
RFC	Request For Comment
SDN	Specification Development Note
UML	Unified Modeling Language
URL	Uniform Resource Locator
WSDL	Web Services Description Language

## 1.6 References

- [APG, 05a] *IMS GLC Application Profile Guidelines Overview: Part 1 – Management Overview v1.0*, IMS Global Learning Consortium, K.Riley, October 2005. <http://www.imsglobal.org/ap/index.html>.
- [APG, 05b] *IMS GLC Application Profile Guidelines White Paper: Part 2 Technical Manual*, S.Wilson and K.Riley, Version 1.0, IMS Global Learning Consortium, October 2005. <http://www.imsglobal.org/ap/index.html>.
- [BDEMS, 11] *IMS GLC Bulk Data Exchange Management Service v1.0 Information Model v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [GWS, 05] *IMS GLC General Web Services WSDL Binding Guidelines v1.0 Final Specification*, C.Schroeder, J.Simon and C.Smythe, IMS Global Learning Consortium, December 2005.
- [IAF, 03a] *IMS GLC Abstract Framework: Applications, Services & Components v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.
- [IAF, 03b] *IMS GLC Abstract Framework: Glossary v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.
- [IAF, 03c] *IMS GLC Abstract Framework: White Paper v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.

- [LIP, 01] *IMS GLC Learner Information Package Information Model v1.0 Final Release*, R.Robson, F.Tansey and C.Smythe, IMS Global Learning Consortium, March 2001.
- [LIS, 11a] *IMS GLC Learner Information Services v2.0 Overview v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [LIS, 11b] *IMS GLC Learner Information Services v2.0 Specification v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [LIS, 11c] *IMS GLC Learner Information Services v2.0 Best Practices & Implementation Guide v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [PMS, 11] *IMS GLC Person Management Service v2.0 WSDL Binding v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [SDN07, 06] *IMS GLC Specification Note 7: UML Profile for Platform Independent Model Descriptions of Specifications for Data Models v1.0*, C.Smythe, IMS Global Learning Consortium, October 2006.
- [SDN11, 06] *IMS GLC Specification Note 11: Vocabulary Definition, Registration & Maintenance Procedures*, C.Smythe, IMS Global Learning Consortium, October 2006.
- [VDEX, 04] *IMS Vocabulary Definition Exchange Best Practice and Implementation Guide, Version 1.0 Final Specification*, A. Cooper, IMS Global Learning Consortium, 2005. Online version:  
[http://www.imsglobal.org/vdex/vdexv1p0/imsvdex\\_bestv1p0.html](http://www.imsglobal.org/vdex/vdexv1p0/imsvdex_bestv1p0.html)

## 2 Person Management Services Description

### 2.1 An Abstract Representation

It is important to remember that this document contains a description of the underlying information model in terms of the abstract API. The manner in which this abstract representation is visualized is not intended to dictate the implementation form of a Person Management Service. The breakdown of the service into its interface classes is a convenient way to document the set of behaviors. The objective for producing these interfaces is to identify and define the messages that are exchanged between the end-systems to realize the system behaviors required of the service.

The internal organization of an implementation of the full abstract API is beyond the scope of this specification. The only constraint is that the external behavior of the abstract API complies with this specification. This means that a .NET, J2EE, etc. physical implementation of this abstract API does not have to represent the functionality using the same breakdown of operations/methods. This physical implementation is not subject to the conformance specification.

It is important to note that the UML representation of the interfaces is used to help develop and document the Person Management Services Information Model. It is not a requirement for a system to implement this interface as defined i.e., to use the same parameters, etc. Conformance against this specification will be confirmed by inspecting the appropriate binding of the information model and ensuring that the relevant information is present and that different sequences of activity result in the predicted and mandated behavior. It is essential that the behaviors described by each of the operations are fully supported and that the behaviors described by different sequences are also maintained.

### 2.2 Person Management Service Architecture & Specification Model

The basic architectural model for the Person Management Service specification is shown in Figure 2.1. In this architecture the scope of the IMS GLC Person Management Service specification is shown as the dotted line. The scope of the interoperability is the data and behavioral models of the objects being exchanged.

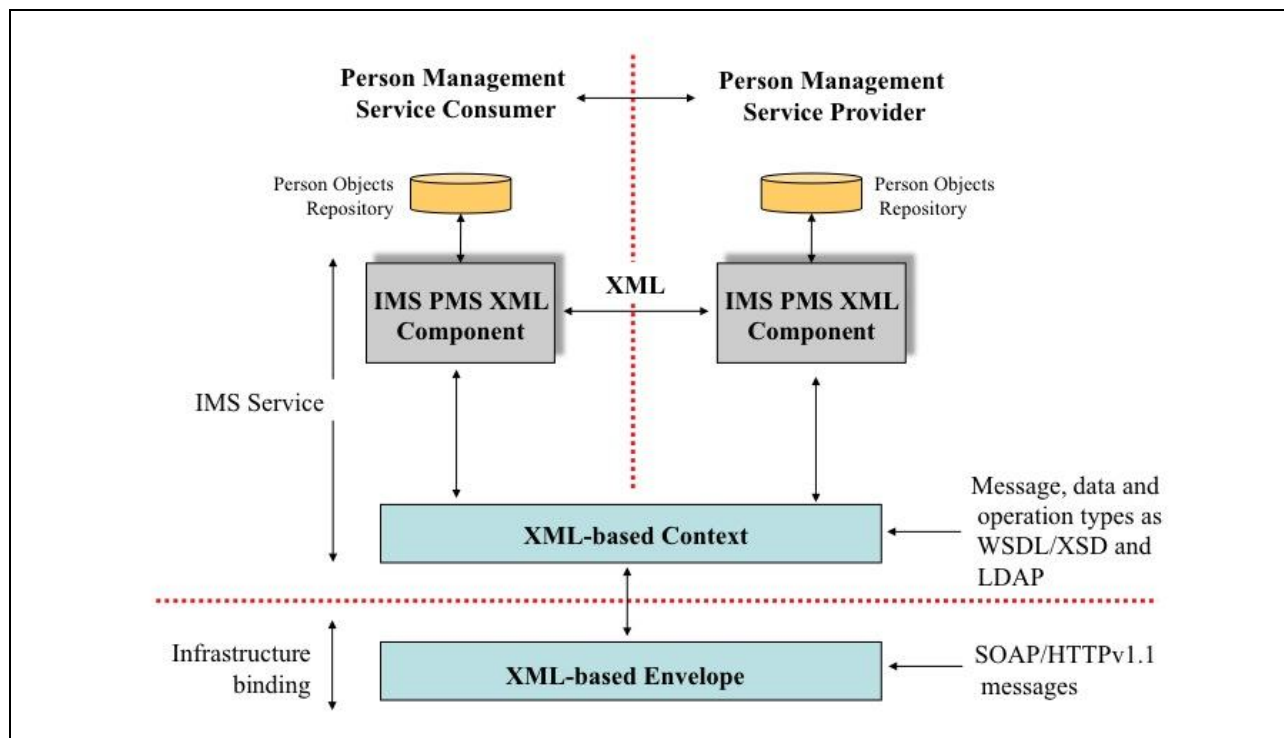


Figure 2.1 Person management service architecture model.

It is important to remember that the structure of the exchanged information has NO bearing on how the same information is contained within the 'source' and 'target' LIS systems (the Person object repositories in the two end-systems). It is simply a representation of the data used to facilitate exchange between the end-systems. The only constraint on the end-system repositories is that they provide data persistence consistent with the required behavior.

## 2.3 Person Objects

It is important to note that this is an **interoperability** specification and as such it makes no statements about how information is stored within the exchanging end systems. The objects in the end-systems **must** be persistent otherwise some sequences of operation on the same object will not be possible. Reference to these objects in the interface is through a 'sourcedId' however this identifier does not have to be the key stored within the end-systems. If different keys are used in the end-systems then it is the responsibility of the end-systems to maintain the mapping between that key and the 'sourcedId' i.e., the interface must never be exposed to the internal keys of the end-systems.

## 2.4 Synchronous & Asynchronous Services

Within the context of the Person Management Service the definition of synchronous and asynchronous services is:

- Synchronous – the source service is blocked until the final response from the target service is received. A schematic representation of the information flow for a synchronous service is shown in Figure 2.2;
- Asynchronous – the source service is not blocked and so more than one request can be outstanding at any moment in time. A schematic representation of the information flow for an asynchronous service is shown in Figure 2.3.

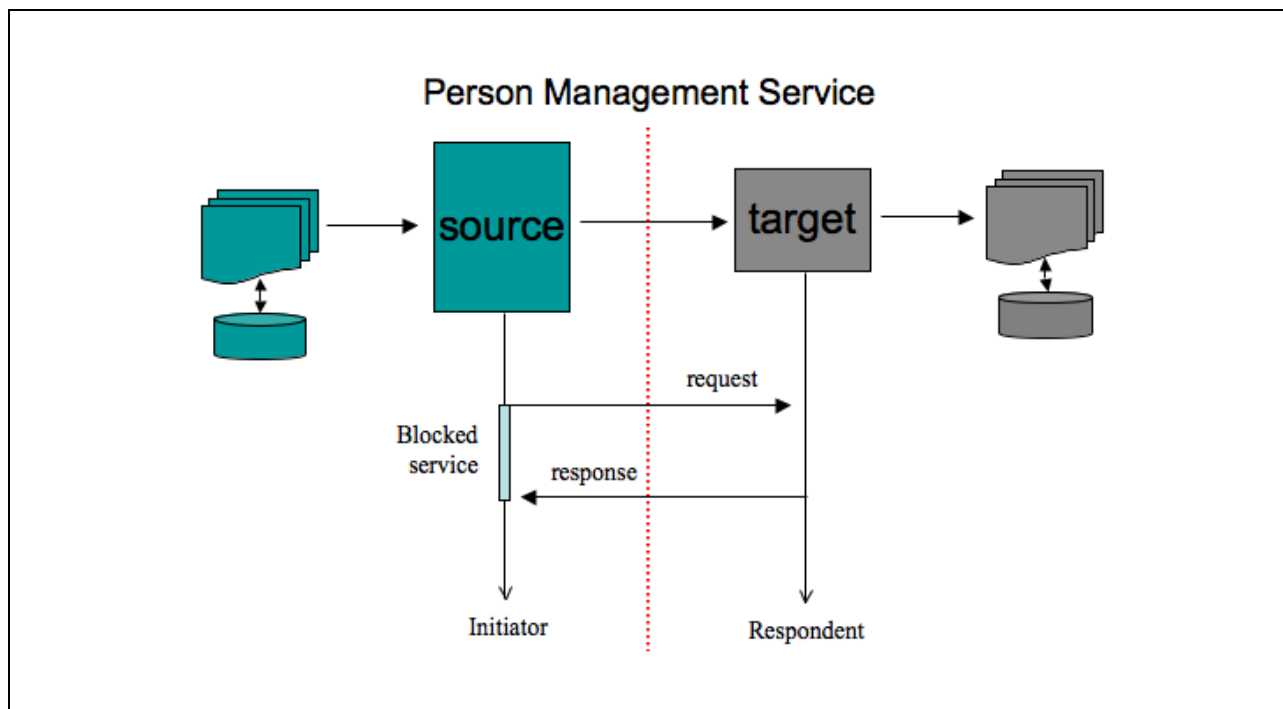
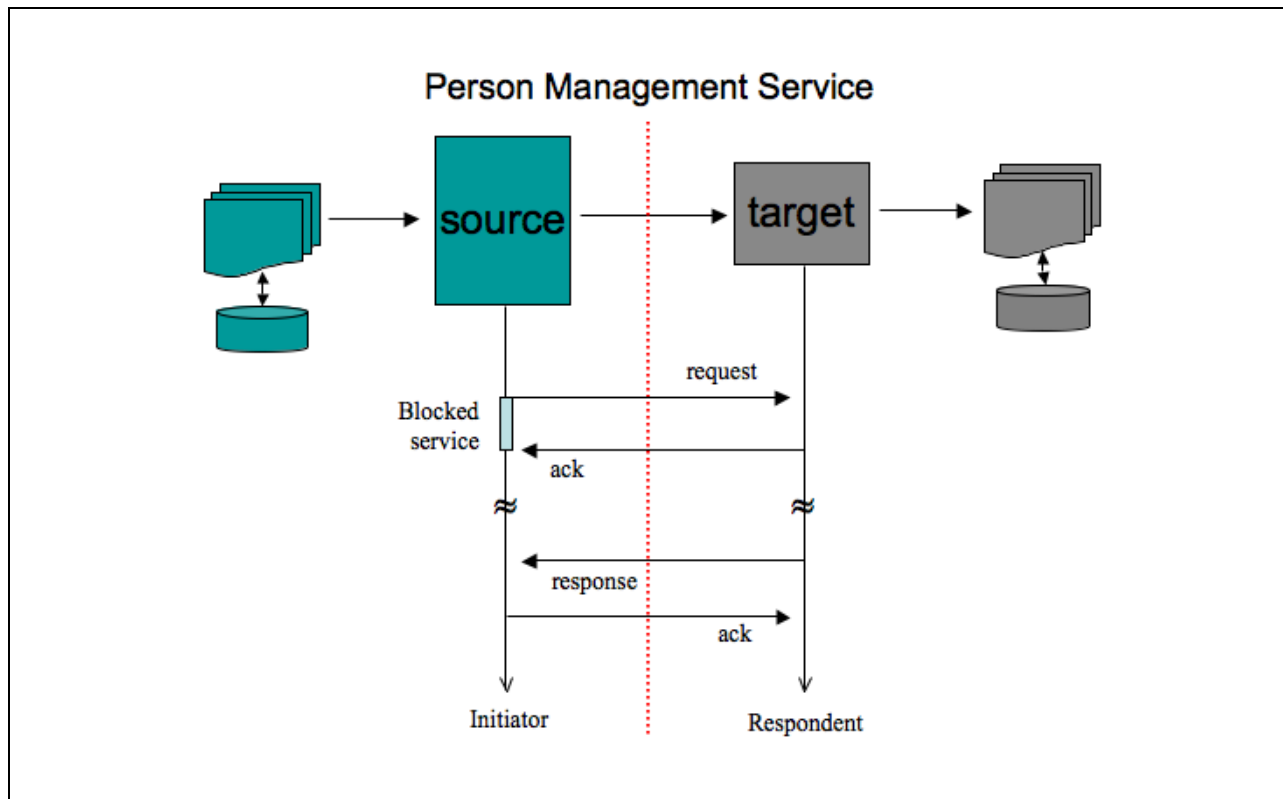


Figure 2.2 Synchronous service.

It is stressed that the abstract-API does not differentiate between synchronous and asynchronous services<sup>1</sup>. The support for these two approaches is provided by different bindings i.e., the same Information Model is used.

<sup>1</sup> In many implementations of the abstract-API the synchronous and asynchronous services would require different operation calls. This is just one example where an implementation does not need to match the definition of the abstract-API.



**Figure 2.3 Asynchronous service.**

The key difference is that for an asynchronous service more than one request can be issued at any one time (it should be noted that an asynchronous service can be supported using synchronous messaging). In both cases the service assumes a perfect messaging system i.e., request, response and acknowledgement messages have a guaranteed delivery grade of service.

## 2.5 Handling the Service Status Codes

Each operation in a service is mapped to an appropriate message exchange pattern. Any response/acknowledgement message will contain status information. This status information provides contextual information about the completed success or otherwise of the operation. There are two types of status information that are available to the end-systems:

- **Business transaction** – these are the status reports that reflect the business logic of the transactions being exchanged by the end-systems. This status information will be contained within the message header under a specially defined data structure. The status information contained herein is also used to contain any error codes i.e., error reporting is handled as a subset of status information reporting;
- **Messaging fault**– these are fault codes that are reported by the messaging infrastructure and which are carried in the messages.

It is important to note that messaging errors may indicate that the original request never reached the service provider end-system. In this case the service consumer implementation that handles the status information is responsible for mapping the message infrastructure failure codes to the equivalent business transaction status code. The message infrastructure failure codes have no meaning with respect to an IMS GLC specification. The IMS GLC specifications do not describe how the status information is to be handled within an end-system i.e., this will depend on how the abstract API is physically realized within an implementation. Therefore, it is important that an implementation can:



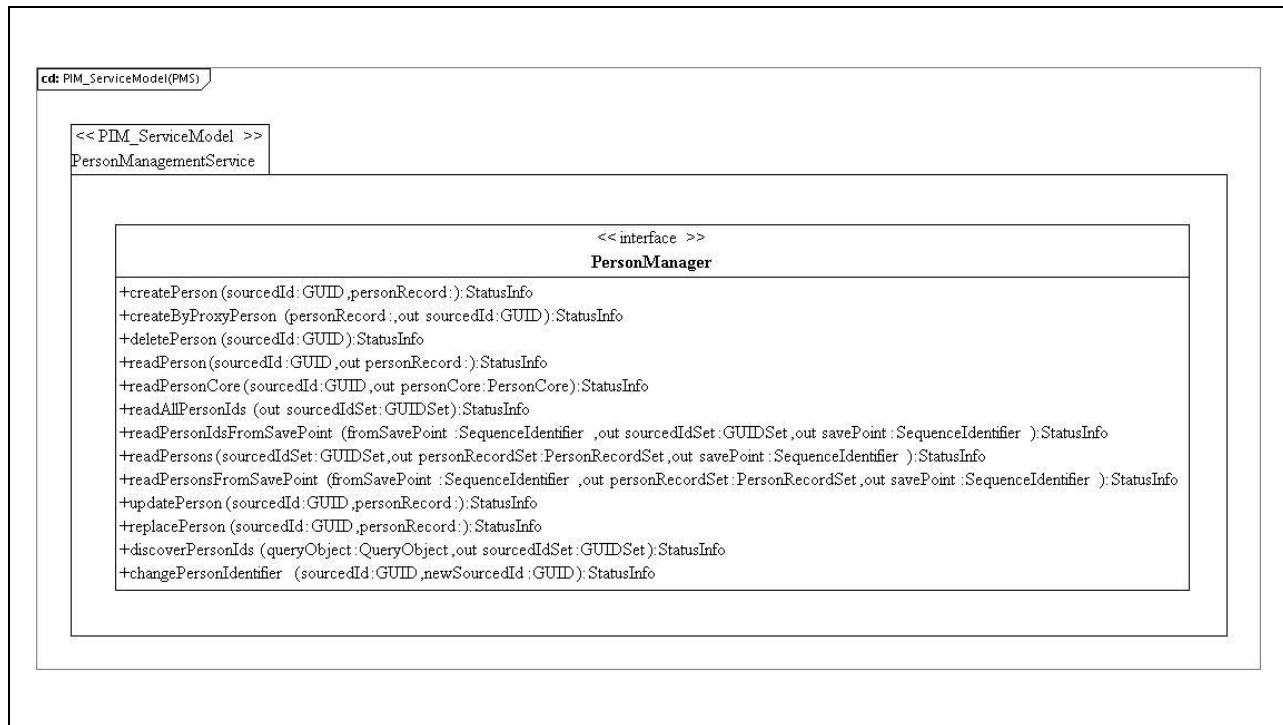
- Combine the transaction status information and any message fault error codes in a single integrated status reporting mechanism. Any other system failure information that is made available by an implementation should also use the same mechanism;
- Examine the status information reported after the completion of the appropriate phase of an operation and especially once the operation has been completed. This may require an explicit status information call or it may be reported as part of the API call;
- Differentiate the status information reports for each transaction within an operation. Remember that some specifications provide operations that can contain more than one transaction request and that a different status report may be given for each of those transactions.

Exception handling is the system's response to known or unknown error conditions. Exception handling is outside the scope of an IMS GLC specification. However, an error condition should not cause the end-systems to fail in an uncontrolled manner. The requirement for every operation to return status information is to enable an implementation to terminate in a controlled fashion.

## 3 Behavioral Model

### 3.1 Service Definition

The PersonManagementService is used to model the service responsible for manipulating information about people. The PersonManagementService interface is shown in Figure 3.1.



**Figure 3.1 PersonManagementService interface definition.**

The PersonManagementService has a single interface: PersonManager that supports the manipulation of Person objects.

### 3.2 PersonManager Interface Description

The PersonManager interface class describes the operations that are permitted on Person objects. These operations are based upon the classic Create/Read/Update/Delete model with variations defined to differentiate subtleties of functionality. The interface stereotype indicates that there are no attributes for this class. The set of operations are summarized in Table 3.1.

**Table 3.1 Summary of operations for PersonManager.**

Operation	Description
createPerson	To request the creation of a populated Person object on the target system where the source is responsible for the allocation of the unique identifier.
createByProxyPerson	To request the creation of a populated Person object on the target system where the target is responsible for the allocation of the unique identifier.
deletePerson	To request the deletion of a Person object. The Person object is deleted and all of its associated memberships.
readPerson	To read the full contents of the identified Person object. The target must return all of the data it has for the identified Person object.
readPersonCore	To read the minimal mandatory contents of the identified Person object. Only the data structures that form the core of a Person object must be returned.
readAllPersonIds	To obtain the set of identifiers which have been assigned to Person objects.
readPersonIdsFromSavePoint	To obtain the set of identifiers for Person objects which have been altered since the requested reference point. The reference point is set as 'zero' at creation and incremented after every write operation.
readPersons	To obtain the Person objects for a defined set of identifiers. This results in a single transaction that may require the exchange of a large volume of data in the response message.
readPersonsFromSavePoint	To obtain the set of Person objects which have been altered since the requested reference point. The reference point is set as 'zero' at creation and incremented after every write operation. This results in a single transaction that may require the exchange of a large volume of data in the response message.
updatePerson	To write new content into the identified Person object. The target must write the new data into the Person object. This is an additive operation.
replacePerson	To replace the content of the identified Person object. The target must write the new data into the Person object. This is a destructive write-over of all of the original information. In the case of the object not existing, this operation acts as an implied 'createPerson'.
discoverPersonIds	To obtain the set of identifiers for Person objects whose properties agree with those defined in the query/filter.
changePersonIdentifier	To change the sourcedId of the Person object. The completion of this operation will result in later actions using the original sourcedId reporting an unknown identifier status.

Note: In most cases the above operations act on a single instance of a Person object i.e., 'createPerson', 'createByProxyPerson', 'deletePerson', 'readPerson', 'replacePerson' and 'updatePerson'.

### 3.2.1 CreatePerson() Operation

<b>Name:</b>	createPerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the creation request. The permitted status codes are defined in Tables 3.2 and A.2.
<b>Supplied (in) Parameters:</b>	<p><i>sourcedId:GUID</i> – the sourcedId allocated by the source system. This is the identifier that must also be assigned within the target system.</p> <p><i>personRecord:PersonRecord</i> – the Person data that is to be stored in the new object.</p>
<b>Returned (out) Parameters:</b>	None.
<b>Behavior:</b>	When the source issues the ‘createPerson’ request the target is instructed to create the populated Person object and to allocate that structure the ‘sourcedId’ passed by the source. If the supplied ‘sourcedId’ has already been allocated to another object then the request is rejected and the appropriate failure code is returned. The save-point identifier is set to ‘zero’ <sup>2</sup> for the Person object in both the source and target.
<b>Notes:</b>	This request contains the initial content for the Person object. More content can be added/replaced using the ‘updatePerson’ and/or ‘replacePerson’ requests.

**Table 3.2 Status codes for the ‘createPerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The creation request has been fully and successfully implemented by the target system and the Person object has been created with a unique identifier.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=idallocinusefail’	The target could not allocate the required unique ‘identifier’ to the Person object as it is already in use.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=overflowfail’	The target could not create the Person object due to lack of target allocation memory.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the supplied data was detected as invalid by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’	The target system could not identify the defined metadata vocabulary

<sup>2</sup> The form taken by ‘zero’ is defined in Section 4.6.

Status Code	Explanation of the Cause of the Code
'Severity=Status' 'CodeMinor=unknownmdvocabulary'	term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownextension'	The target cannot process the proprietary data model extensions used in the object.

### 3.2.2 CreateByProxyPerson() Operation

<b>Name:</b>	createByProxyPerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the creation request. The permitted status codes are defined in Tables 3.3 and A.2.
<b>Supplied (in) Parameters:</b>	<i>personRecord:PersonRecord</i> – the Person data that is to be stored in the new object.
<b>Returned (out) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier allocated by the target to the newly created ‘person’ object.
<b>Behavior:</b>	When the source issues the ‘createByProxyPerson’ request the target is instructed to create the populated Person object and to allocate a unique ‘identifier’. The save-point identifier is set to ‘zero’ for the ‘person’ object in both the source and target.
<b>Notes:</b>	This request contains the initial content for the Person object. More content can be added/replaced using the ‘updatePerson’ and/or ‘replacePerson’ requests.

**Table 3.3 Status codes for the ‘createByProxyPerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The creation request has been fully and successfully implemented by the target system and the Person object has been created with the identifier supplied by the source.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=idallocfail’	The target could not allocate a unique ‘identifier’ to the Person object because there are no more spare identifiers available.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=overflowfail’	The target could not create the Person object due to lack of target allocation memory.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the supplied data was detected as invalid by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’	The target cannot process the proprietary data model extensions used in

Status Code	Explanation of the Cause of the Code
'Severity=Status' 'CodeMinor=unknownextension'	the object.

### 3.2.3 DeletePerson() Operation

<b>Name:</b>	deletePerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the creation request. The permitted status codes are defined in Tables 3.4 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier to be used by the target to identify the Person object.
<b>Returned (out) Parameters:</b>	None.
<b>Behavior:</b>	When the source issues the ‘deletePerson’ request the target is instructed to delete the identified Person object and to remove the reference to the Person from any of the related Membership objects. This is a hard cascaded delete from which there is no recovery. If the object identified by the supplied ‘sourcedId’ cannot be located then the request is rejected and the appropriate failure code is returned.
<b>Notes:</b>	Deletion of the Person record does not necessarily result in the destruction of the data in the server. The real state of the data in the target is unknown.

**Table 3.4 Status codes for the ‘deletePerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The deletion request has been fully and successfully implemented by the target system and the Person object has been deleted. The corresponding membership records have also been deleted.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The Person object identifier is unknown in the target system and so the object could not be deleted.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor= deletefailure’	The target system has not been able to delete the identified Person object.



### 3.2.4 ReadPerson() Operation

<b>Name:</b>	readPerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.5 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier of the Person object to be read.
<b>Returned (out) Parameters:</b>	<i>personRecord:PersonRecord</i> – the Person data that is read from the object.
<b>Behavior:</b>	When the source issues the ‘readPerson’ request the target is charged with retrieving the identified record from its database and returning this data to the source. The target is responsible for ensuring that the record contains valid data. If the object identified by the supplied ‘sourcedId’ cannot be located then the request is rejected and the appropriate failure code is returned.
<b>Notes:</b>	The returned Person record can only be trusted if the corresponding status code is ‘success’.

**Table 3.5 Status codes for the ‘readPerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the identified Person object has been read from the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The Person object identifier is unknown in the target system and so the object could not be read.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=targetreadfailure’	The target system has detected an error in the stored Person object and so cannot return the data.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the source system.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has only returned a subset of the data expected by the source e.g., only the mandatory parts or only some of the possible set of optional entries.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The source cannot process the proprietary data model extensions used in the object.

### 3.2.5 ReadCorePerson() Operation

<b>Name:</b>	readCorePerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.6 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier of the Person object to be read.
<b>Returned (out) Parameters:</b>	<i>personCore:PersonCore</i> – the core, mandatory, Person data that is read from the object.
<b>Behavior:</b>	When the source issues the ‘readCorePerson’ request the target is charged with retrieving the mandatory core data structures for the identified record from its database and returning this data to the source. The target is responsible for ensuring that the record contains valid data. If the object identified by the supplied ‘sourcedId’ cannot be located then the request is rejected and the appropriate failure code is returned. If any mandatory field is missing then the status returned is ‘incompletedata’.
<b>Notes:</b>	The returned Person record can only be trusted if the corresponding status code is ‘success’.

**Table 3.6 Status codes for the ‘readCorePerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the identified Person object data has been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The Person object identifier is unknown in the target system and so the object could not be read.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=targetreadfailure’	The target system has detected an error in the stored Person object and so cannot return the data.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the source system.

### 3.2.6 ReadAllPersonIds() Operation

<b>Name:</b>	readAllPersonIds
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.7 and A.2.
<b>Supplied (in) Parameters:</b>	None.
<b>Returned (out) Parameters:</b>	<i>sourcedIdSet:GUIDSet</i> – the set of identifiers of the Person objects stored on the target.
<b>Behavior:</b>	When the source issues the ‘readAllPersonIds’ the target returns the set of sourcedIds that have been allocated to Person objects.
<b>Notes:</b>	If no sourcedIds have been allocated then the returned data set is empty and the success status code returned.

**Table 3.7 Status codes for the ‘readAllPersonIds’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the Person object identifiers have been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor= nosourcedids’	The read request has been fully and successfully implemented by the target system and no Person object identifiers were found.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the source system.

### 3.2.7 ReadPersonIdsFromSavePoint() Operation

<b>Name:</b>	readPersonIdsFromSavePoint
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.8 and A.2.
<b>Supplied (in) Parameters:</b>	<i>fromSavePoint:SequenceIdentifier</i> – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
<b>Returned (out) Parameters:</b>	<i>sourcedIdSet:GUIDSet</i> – the set of identifiers of the Person objects stored on the target.  <i>savePoint:SequenceIdentifier</i> – the value of the reference point counter in the target system.
<b>Behavior:</b>	When the source issues the ‘readGroupIdsFromSavePoint’ the target returns the set of SourcedIds that have been altered from the defined reference point, supplied by the source, to the latest reference point identified in the target.  If the reference counter in the source is greater than that in the target then an error code is returned along with the target value for the reference point.
<b>Notes:</b>	If the target has no SourcedIds assigned to Person objects or there have been no changes in the defined period then the returned data set is empty and the success status code returned.

**Table 3.8 Status codes for the ‘readPersonIdsFromSavePoint’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the Person object identifiers have been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor= nosourcedids’	The read request has been fully and successfully implemented by the target system and no Person object identifiers were found.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointsyncerror’	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value is reported to the source system for information.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointerror’	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.

### 3.2.8 ReadPersons() Operation

<b>Name:</b>	readPersons
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.9 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedIdSet:GUIDSet</i> – the set of identifiers of the Person objects to be read.
<b>Returned (out) Parameters:</b>	<i>personRecordSet:PersonRecordSet</i> – the set of person records. <i>savePoint:SequenceIdentifier</i> – the value of the reference point counter in the target system.
<b>Behavior:</b>	When the source issues the ‘readPersons’ request the target is charged with retrieving the identified set of objects from its database. The associated read savePoint reference is updated and returned.  If the object identified by the supplied SourcedId cannot be located then the request is rejected and a partial success code is returned for the operation. The target is responsible for ensuring that the records contain valid data. The target should attempt to successfully complete as much of the request as possible.
<b>Notes:</b>	A returned Person record is only present if the object has been located in the target system and the full data set returned.  The enclosed data may result in a long response message.

**Table 3.9 Status codes for the ‘readPersons’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the identified Person object has been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=partialreadfail’	Some of the Person object identifiers are unknown in the target system and so those objects could not be read.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.

### 3.2.9 ReadPersonsFromSavePoint() Operation

<b>Name:</b>	readPersonsFromSavePoint
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the read request. The permitted status codes are given in Tables 3.10 and A.2.
<b>Supplied (in) Parameters:</b>	<i>fromSavePoint:SequenceIdentifier</i> – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
<b>Returned (out) Parameters:</b>	<i>personRecordSet:PersonRecordSet</i> – the set of person records. <i>savePoint:SequenceIdentifier</i> – the value of the reference point counter in the target system.
<b>Behavior:</b>	When the source issues the ‘readPersonsFromSavePoint’ request the target is charged with reading the objects that have been altered from the defined reference point.  If the reference counter in the source is greater than that in the target then an empty data file is returned and the target value for the reference point is returned.
<b>Notes:</b>	If no objects have been allocated then the return message will be empty.  The enclosed data may result in a long response message.

**Table 3.10 Status codes for the ‘readPersonsFromSavePoint’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the Person objects have been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=partialreadfail’	Some of the Group object identifiers are unknown in the target system and so those objects could not be read.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointerror’	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointsyncerror’	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value is reported to the source system for information.

### 3.2.10 UpdatePerson() Operation

<b>Name:</b>	updatePerson
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the update request. The permitted status codes are given in Tables 3.11 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier of the Person object to be updated. <i>personRecord:PersonRecord</i> – the Person data to be stored in the object.
<b>Returned (out) Parameters:</b>	None.
<b>Behavior:</b>	<p>When the source issues the ‘updatePerson’ request the target is charged with writing the supplied information into the identified object. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is an additive write operation of all the data fields supplied in the update request and fields not supplied remain unchanged. If a field is constrained with a multiplicity of one the ‘updatePerson’ request acts as a replacePerson’ request for that field.</p> <p>If the object identified by the supplied ‘SourcedId’ cannot be located then the request is rejected and the appropriate failure code is returned.</p> <p>The reference pointer for the object is incremented in the target system.</p>
<b>Notes:</b>	The source is responsible for determining the reason of the failure.

**Table 3.11 Status codes for the ‘updatePerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The update request has been fully and successfully implemented by the target system and the identified Person object has been changed on the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The Person object identifier is unknown in the target system and so the object could not be updated.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has only stored a subset of the sent data object e.g., only the mandatory parts.

<p>'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownextension'</p>	<p>The target cannot process the proprietary data model extensions used in the object.</p>
---	--



### 3.2.11 ReplacePerson() Operation

<b>Name:</b>	replacePerson
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the replace request. The permitted status codes are given in Tables 3.12 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier of the Person object to be replaced. <i>personRecord:PersonRecord</i> – the Person data that is to be stored in the object.
<b>Returned (out) Parameters:</b>	None.
<b>Behavior:</b>	<p>When the source issues the ‘replacePerson’ request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is a destructive write-over operation of the entire ‘person’ object. This is equivalent to a ‘createPerson’ but for an object that already exists.</p> <p>If the object identified by the supplied SourcedId cannot be located then the request is interpreted as a ‘createPerson’ invocation.</p> <p>The reference pointer for the object is incremented in the target system.</p>
<b>Notes:</b>	The source is responsible for determining the reason of the failure.

**Table 3.12 Status codes for the ‘replacePerson’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The replace request has been fully and successfully implemented by the target system and the identified Person object has been changed on the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=createsuccess’	The Person object identifier is unknown in the target system and so a new object has been successfully created instead.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined <i>metadata</i> vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownextension'	The target cannot process the proprietary data model extensions used in the object.

### 3.2.12 DiscoverPersonIds() Operation

<b>Name:</b>	discoverPersonIds
<b>Return Function Parameter:</b>	<i>statusInfo:StatusInfo</i> – the status of the discover request. The permitted status codes are given in Tables 3.13 and A.2.
<b>Supplied (in) Parameters:</b>	<i>queryObject:QueryObject</i> – this is the query/filter instruction that is to be applied by the target to discover the corresponding Person objects.
<b>Returned (out) Parameters:</b>	<i>sourcedIdSet:GUIDSet</i> – the set of identifiers of the Person objects whose content conform to the query/filter conditions.
<b>Behavior:</b>	<p>When the source issues the ‘discoverPersonIds’ the target applies the query/filter instructions to the set of Person objects and returns the set of sourcedIds that uphold the query.</p> <p>If no Person objects have the required properties the returned data set is empty and the success status code returned.</p> <p>If the target does not understand or cannot apply the requested query/filter then an error status is returned.</p>
<b>Notes:</b>	The internal structure of this QueryObject is undefined (it is should be treated as a String encoded 'blob'). Later versions of this specification will look at the established best practices for clarification on the use of this operation.

**Table 3.13 Status codes for the ‘discoverPersonIds’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The discover request has been fully and successfully implemented by the target system and the appropriate Person identifiers have been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor= nosourcedids’	The discover request has been fully and successfully implemented by the target system and no Person object identifiers were found.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownquery’	The target system cannot understand the query request that has been received i.e., the query/filter language is unknown.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the supplied data was detected as invalid by the source system.

### 3.2.13 ChangePersonIdentifier() Operation

<b>Name:</b>	changePersonIdentifier
<b>Return Function Parameter:</b>	<i>StatusInfo</i> – the status of the update request. The permitted status codes are given in Tables 3.14 and A.2.
<b>Supplied (in) Parameters:</b>	<i>sourcedId:GUID</i> – the identifier of the Person object to be changed. <i>newSourcedId:GUID</i> – the new identifier to be allocated to the Person object.
<b>Returned (out) Parameters:</b>	None.
<b>Behavior:</b>	When the source issues the ‘changePersonIdentifier’ request the target is charged with replacing the original ‘sourcedId’ with the new supplied ‘sourcedId’. All membership entries must be similarly changed. All further references to the object must use the new ‘sourcedId’ otherwise an ‘unknown’ object failure status code is returned.  If the object identified by the supplied ‘SourcedId’ cannot be located then the request is rejected and the appropriate failure code is returned.
<b>Notes:</b>	The reference pointer value remains unchanged.

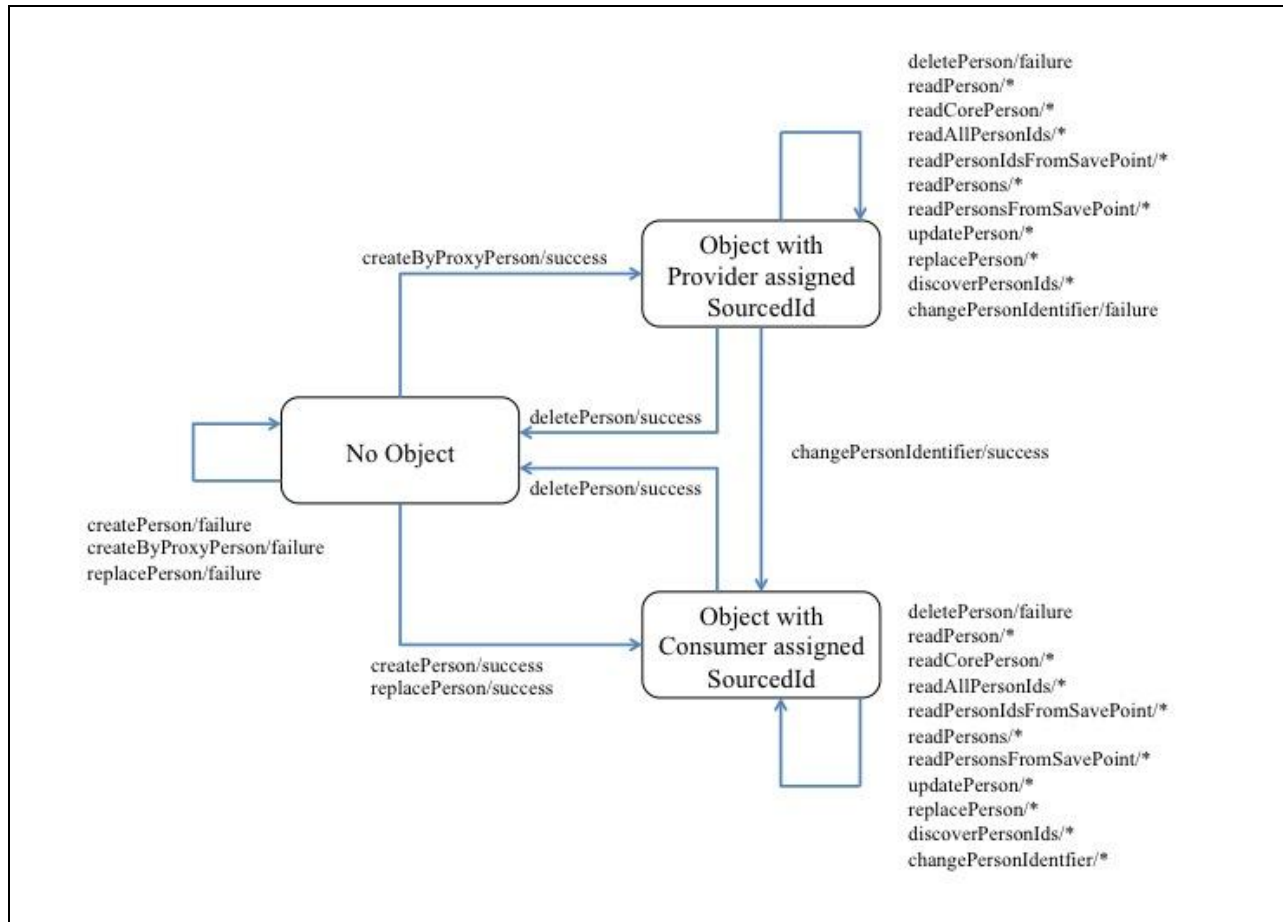
**Table 3.14 Status codes for the ‘changePersonIdentifier’ operation.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The change identifier request has been fully and successfully implemented by the target system and the Person object ‘sourcedId’ has been changed on the target system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=idallocinusefail’	The target could not allocate the new unique ‘identifier’ to the Person object as the identifier is already in use.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The current ‘sourcedId’ identifier is unknown in the target system and so the object identifier could not be changed.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownobject’	The current ‘sourcedId’ identifier is unknown in the target system and so the object identifier could not be changed.

### 3.3 Person Object State Machine

The permitted state activity on a Person object is shown in Figure 3.2. This state diagram has three states (the arcs are annotated with the operations that are associated with the change of state):

- ‘No Object’ state – no Person object exists with a particular sourcedId;
- ‘Object with Provider assigned sourcedId’ – a Person object exists with the sourcedId allocated by the Provider system;
- ‘Object with Consumer assigned sourcedId’ – a Person object exists with the sourcedId allocated by the Consumer system.



**Figure 3.2 State machine for a ‘person’ object.**

The start state is ‘No Object’ i.e., the Person object has not yet been created. Only the ‘createPerson()’, ‘replacePerson’ and ‘createByProxyPerson()’ operations are possible. Once the Person object has been created then it persists until a successful ‘deletePerson()’ operation is completed. The ‘createPerson()’ and ‘replacePerson’ operations take the system into the ‘Object with Consumer assigned sourcedId’ state whereas the ‘createByProxyPerson()’ takes the system into the ‘Object with Provider assigned sourcedId’ state.

The system can be moved from the ‘Object with Provider assigned sourcedId’ state into the ‘Object with Consumer assigned sourcedId’ state by the successful completion of the ‘changePersonIdentifier()’ operation.

Once the system is in the ‘Object with Consumer assigned sourcedId’ or the ‘Object with Provider assigned sourcedId’ states then the ‘readPerson()’, ‘readCorePerson()’, ‘readAllPersonIds()’,

‘readPersonIdsFromSavePoint()’, ‘readPersons()’, ‘readPersonsFromSavePoint()’, ‘updatePerson()’, ‘replacePerson()’ and ‘discoverPersonIds()’ operations are now possible.

This is the state machine for each Person object in the Service Consumer and the Service Provider. The binding of the Information Model must guarantee that these two state machines remain synchronized for each Person object.

## 4 Interface Data Model

The set of operations described within the behavior model (Section 3) are based upon class descriptions specific to the parameters of the operations. All parameters are mandatory.

### 4.1 GUID Class Description

This is the data type for the globally unique sourcedIds. These GUIDs must be unique across the set of communicating end-systems within the LIS system. The internal format of the GUID is outside the scope of this specification but they must all be valid strings. Any implementation of the GUID class must be able to support GUIDs of at least 1024 bytes in length i.e., the shortest permitted maximum length.

### 4.2 GUIDSet Class Description

This is the data-type for a set of GUIDs (zero or more). Any implementation of the GUIDSet must be able to contain at least 250,000 GUIDs i.e., the smallest permitted maximum number.

### 4.3 PersonRecord Class Description

This is the data-type for Person objects. The data model for a Person is described in Section 5. A key difference for an object passed in the interface, as opposed to the requirement for an end-system, is that the content is dependent on the type of operation. A PersonRecord object must consist of the SourcedId of the Group object and the Group object itself.

### 4.4 PersonRecordSet Class Description

This is the data-type for a set of PersonRecords (zero or more). Any implementation of the PersonRecordSet must be able to contain at least 250,000 PersonRecords i.e., the smallest permitted maximum number.

### 4.5 PersonCore Class Description

This is the data-type for core, the mandatory subset, of a Person object. Each attribute has a multiplicity of one.

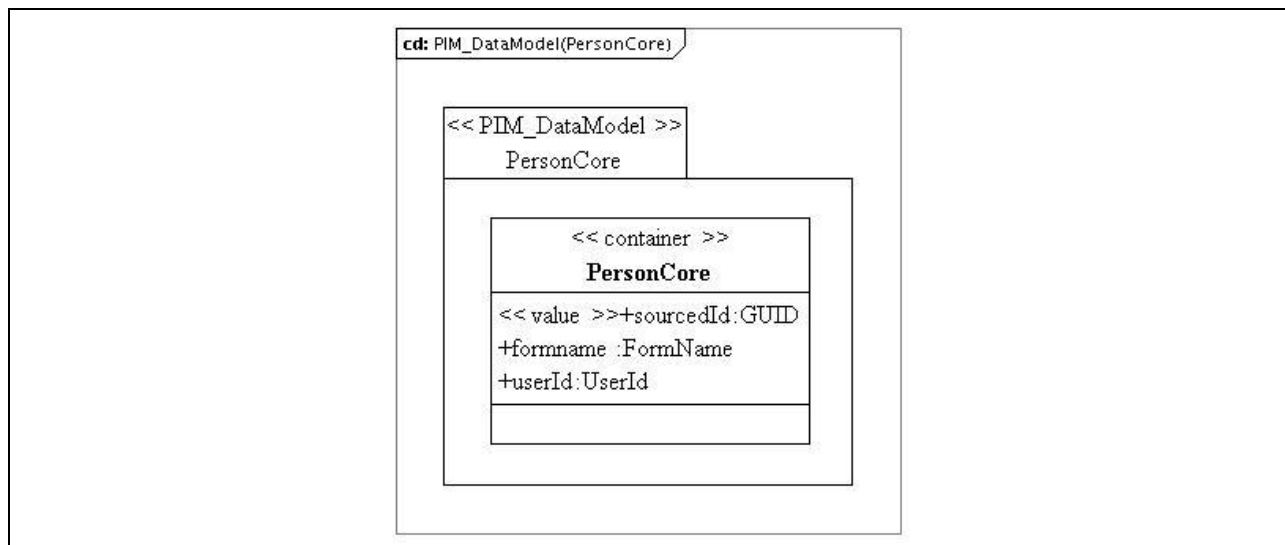


Figure 4.1 The PersonCore class diagram.

This mandatory subset consists of:

- SourcedId – the unique identifier for the Person object;
- FormName – the full name formatted in the way that the Person prefers to be addressed (the structure of this class is described in Sub-section 5.5);

- **UserId** – the user identification information for the Person (the structure of this class is described in sub-section 5.13).

#### 4.5.1 SourcedId Attribute Description

**Table 4.1 Description of the ‘sourcedId’ attribute for the PersonCore class.**

Descriptor	Definition
Attribute name	sourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the globally unique identifier that has been assigned to the associated Person object. Each person object must have only one sourcedId but this may be changed, any number of times, during the object’s lifetime.

#### 4.5.2 Formname Attribute Description

**Table 4.2 Description of the ‘formname’ attribute for the PersonCore class.**

Descriptor	Definition
Attribute name	formname
Data type	FormName (defined in Sub-section 5.5)
Value space	container
Multiplicity	1
Description	Full name formatted in the way that the Person prefers to be addressed.

#### 4.5.3 UserId Attribute Description

**Table 4.3 Description of the ‘userId’ attribute for the PersonCore class.**

Descriptor	Definition
Attribute name	userId
Data type	UserId (defined in Sub-section 5.13)
Value space	container
Multiplicity	1
Description	The person’s user identifier to access the learning management environment.



## 4.6 QueryObject Class Description

This is the data-type for the query instruction. This is a String 'blob' with the smallest permitted maximum length of 4096 octets. The internal structure of this string is undefined. Later versions of this specification will look at the established best practices for clarification on the use of this string.

## 4.7 SequenceIdentifier Class Description

This is the data-type for the sequence identifier used to denote identify the synchronization reference point between the two communicating systems. The sequence is denoted by the date-time string YYYY-MM-DDTHH:MM:SS.NNN where 'YYYY' denotes the year, the first 'MM' string the month (01-12), 'DD' the day (01-31), 'HH' the hour (00-23), the second 'MM' string the minute (00-59), 'SS' the second (00-59) and 'NN' the millisecond value (000-999).

At initialization the value is set to '1000-01-01T00:00:00.000'. The value is changed to the current time for every operation that results in a change of the value of the data stored in the Person object.

All values are to be rounded down at the level of greatest resolution.

## 4.8 StatusInfo Class Description

This is the container for the status information returned by the target to the source. The structure of this class is described in the IMS GLC General Web Services specification v1.0 (Appendix A) [GWS, 05].

## 5 End System Data Model

The end system data model defines the persistence model that must be maintained by an end system to ensure the correct system behavior.

An informative overview of the entire Persistence Data Model is provided as a Platform Independent Model (PIM) expressed in UML constructs. All UML diagrams expressed as “Platform Independent Model” are non-normative. Normative tables defining the classes in this Information Model follow the informative UML diagrams. A full definition of the UML Profile and the terms used in the normative tabular descriptions in this document to describe the PIM can be found in [SDN07, 06].

In the tables in this section the character sequence “n/a” is used to mark a field “not applicable.” Any field so marked is not relevant to the class being defined. Features so marked shall be ignored when binding a class defined by this Information Model.

### 5.1 Key terms and concepts

Classes in this information model are classified into four abstract class types. These abstractions are bound to specific data structures for machine processing in the IMS GLC Person Management Service WSDL Binding Version 2.0 [PMS, 11]. The four abstract class types are:

- **container:** A container class may be a parent of one or more child classes;
- **value:** A value class shall not be a parent. That is, it shall not be a composite of characteristic, container, value, or unspecified class types. A value class shall always be a child of a container class and shall have semantic value within the scope of its parent class’s semantic value;
- **characteristic:** A characteristic class shall not be a parent. A characteristic class shall declare a trait or value that is an intrinsic feature or part of a container class. A characteristic class is tightly coupled to the **container** class it modifies or one of which facets it describes;
- **unspecified: An unspecified class may be a parent.** An unspecified class serves as an extension point for this Information Model.

Table 5.1 lists the class descriptors used to describe the abstract classes and definitions of the descriptors.

**Table 5.1 Class descriptors**

Descriptor	Definition
Class name	The name given to the class being described.
Class type	The abstract class type of this class.
Data type	<p>For value and characteristic classes, the allowed structure for valid values for the class.</p> <p>Valid data types are:</p> <p><b>Boolean:</b> The primitive, two-valued data type that uses the keywords “true” and “false” to indicate the logical state of an object.</p> <p><b>Date:</b> The date represents a date in the format of ISO 8601 i.e., ‘YYYY-MM-DD’.</p> <p><b>DateTime:</b> The DateTime represents a combined date and time in the format of ISO 8601 i.e., ‘YYYY-MM-DDThh:mm:ssTZD’. The time is denoted in Coordinated Universal Time (UTC) with TZD denoting the time zone offset in hours and minutes with respect to UTC.</p> <p><b>GUID:</b> An identifier that is globally unique. This will be based upon the Normalized String data-type that has a constrained value-space. This has a length [1..4095] characters.</p> <p><b>NormalizedString:</b> A sequence of printable characters that does not contain carriage</p>

Descriptor	Definition
	<p>returns or tabs.</p> <p><b>String:</b> A sequence of printable characters.</p> <p><b>Text:</b> A language annotated string (this is in fact a separate class but it is treated as data-type for convenience). The string is accompanied by a language identifier that denotes the language for the string (this code is based upon RFC4646)..</p> <p><b>AnyURI:</b> Any syntactically valid instance of a URI as defined in RFC3986. Note: Many of the foundational Specifications, Standards, and Recommendations referred to by this Information Model use RFC2396 and RFC2732 as the definitions of URI. These are made obsolete by RFC3986, but many of the foundational documents have not been updated to reference RFC3986.</p> <p><b>URL:</b> A normalized string that is used to contain a Universal Resource Location. This has a length [1..4095] characters.</p> <p><b>Unspecified:</b> The data type is not known or is not important.</p>
Value space	The range of valid values for this class. If the value space is unspecified, it is not known or is not important.
Multiplicity	<p>A property of a class indicating the number of times it may be used or appear in a given parent context. The values of this property are expressed as a range or shorthand for a range using this notation:</p> <ul style="list-style-type: none"> <li>• <b>'0..1'</b> [optional; restricted]</li> <li>• <b>'0..unbounded'</b> [optional; unrestricted]</li> <li>• <b>'1..1'</b> [mandatory; restricted]</li> <li>• <b>'1..unbounded'</b> [mandatory; unrestricted]</li> </ul> <p>Multiplicities may also appear in short-hand notation in the UML models. The short-hand equivalents shall be (exclusive of bracketed comments):</p> <ul style="list-style-type: none"> <li>• <b>'*'</b> [optional; unrestricted]</li> <li>• <b>'1'</b> [mandatory; restricted]</li> <li>• <b>'1..*'</b> [mandatory; unrestricted]</li> </ul> <p>Where multiplicity is greater than one, the importance of the ordering of siblings is also indicated by appending either <b>","ordered</b> or <b>"," unordered</b>.</p> <p><b>Ordered</b> specifies a sequence of siblings as listed, <b>unordered</b> specifies a collection or bag of siblings for which the order is not important.</p>
Parents	Lists classes that may be parents of this class.
Children	<p>Lists the possible child classes of this class in the form <b>"[" child *"," child "]"</b>. One or more child classes may be expressed within square brackets. Each child class shall be separated by a comma.</p> <p>Where more than one child is listed, the importance of the ordering of siblings is also indicated by appending either <b>","ordered</b> or <b>"," unordered</b>.</p> <p><b>Ordered</b> specifies a sequence of siblings as listed, <b>unordered</b> specifies a collection or bag of siblings for which the order is not important.</p>
Description	Contains descriptions relating to the class and its values space.

In general, this specification does not define the ways in which an end system must be realized. However, the required interoperability behavior requires that an end system have certain characteristics. The static properties of these characteristics are defined in this Section, including:

- When an attribute has a multiplicity of '1..1' then an end system must be capable of supporting one instance;

- When an attribute has a multiplicity of '1..\*' then an end system must be capable of supporting at least one instance. The specification will also define the smallest permitted maximum number of instances that must also be supported by the end system;
- When an attribute has a multiplicity of '0..1' then an end system should support a single instance;
- When an attribute has a multiplicity of '0..\*' then the specification (in the form of a profile or conformance statement) will define the smallest permitted maximum number of instances that must also be supported by the end system.

When the object is passed as part of a service call then attributes that have a '1..1' or '1..\*' multiplicity may or may not be exchanged. This is because the specification of an end system defines capability; an operational system may or may not exchange the associated information.

## 5.2 PersonDatabase Class Description

The PIM for the PersonDatabase data model is shown in Figure 5.1.

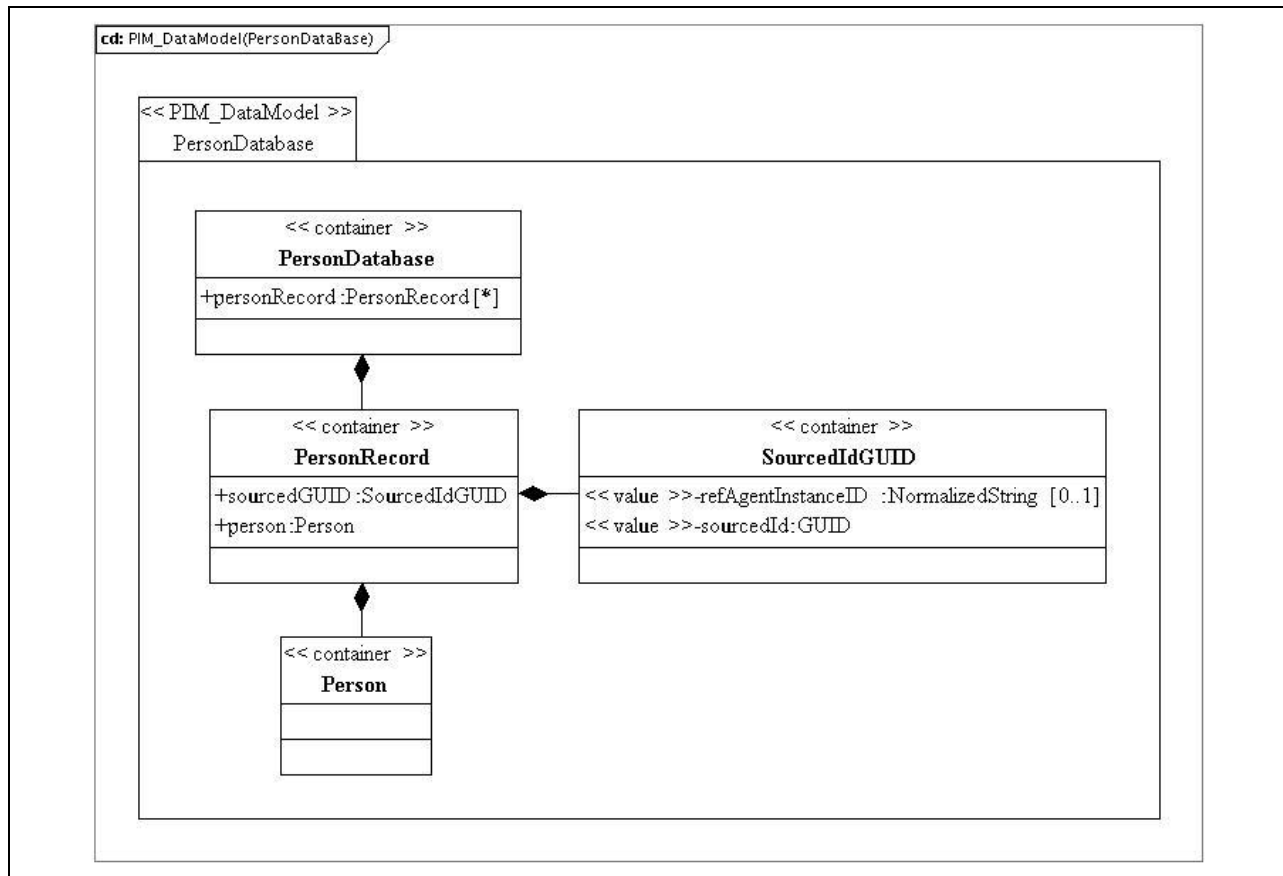


Figure 5.1 PersonDatabase class diagram.

Table 5.2 Description of the 'PersonDatabase' class.

Descriptor	Definition
Class name	PersonDatabase
Class type	container
Multiplicity	1
Parents	Root
Children	[ PersonRecord ]
Description	This is the database within the end-system that contains all of the Person objects. Each Person object consists of a globally unique identifier, its SourcedGUID and the Person data itself. The database consists of the set of Person objects, the set of GUIDs and the relationship mapping between the two. The manner in which this information is physically stored is outside of the scope of this specification.

### 5.2.1 PersonRecord Attribute Description

**Table 5.3 Description of the ‘personRecord’ attribute for the PersonDatabase class.**

Descriptor	Definition
Attribute name	personRecord
Data type	PersonRecord
Value space	container
Multiplicity	0..unbounded, unordered
Description	This is set of PersonRecords that constitute the PersonDatabase. A PersonDatabase must be capable of supporting at least 100,000 PersonRecord instances.

## 5.3 PersonRecord Class Description

**Table 5.4 Description of the ‘PersonRecord’ class.**

Descriptor	Definition
Class name	PersonRecord
Class type	container
Multiplicity	0..unbounded, unordered
Parents	PersonDatabase
Children	[ sourcedGUID, person ], ordered
Description	The person record represents the association of the unique identifier (GUID) for the Person object with the Person object itself. The ‘identifier’ object is not a part of the Person object but both are managed within the Person Database. There is an isomorphic association between each pair of ‘identifier’ and ‘person’ objects.

### 5.3.1 SourcedGUID Attribute Description

**Table 5.5 Description of the ‘sourcedGUID’ attribute for the PersonRecord class.**

Descriptor	Definition
Attribute name	sourcedGUID
Data type	SorcedidGUID
Value space	container
Multiplicity	1
Description	This is the globally unique identifier that has been assigned to the associated Person object. Each Person object must have only one sourcedGUID but this may be changed, any number of times, during the object’s lifetime.

## 5.4 SourcedGUID Class Description

**Table 5.6 Description of the SourcedGUID class.**

Descriptor	Definition
Class name	SourcedGUID
Class type	container
Children	[ refAgentInstanceID, sourcedId ], ordered
Description	This is a structured GUID that consists of an instance identifier and a sourcedId.

### 5.4.1 RefAgentInstanceID Attribute Description

**Table 5.7 Description of the ‘refAgentInstanceID’ attribute for the SourcedGUID class.**

Descriptor	Definition
Attribute name	refAgentInstanceID
Data type	NormalizedString
Value space	Normalized string [1..31 characters].
Multiplicity	0..1
Description	This is an instance identifier used to differentiate, if necessary, between multiple end system reference agents.

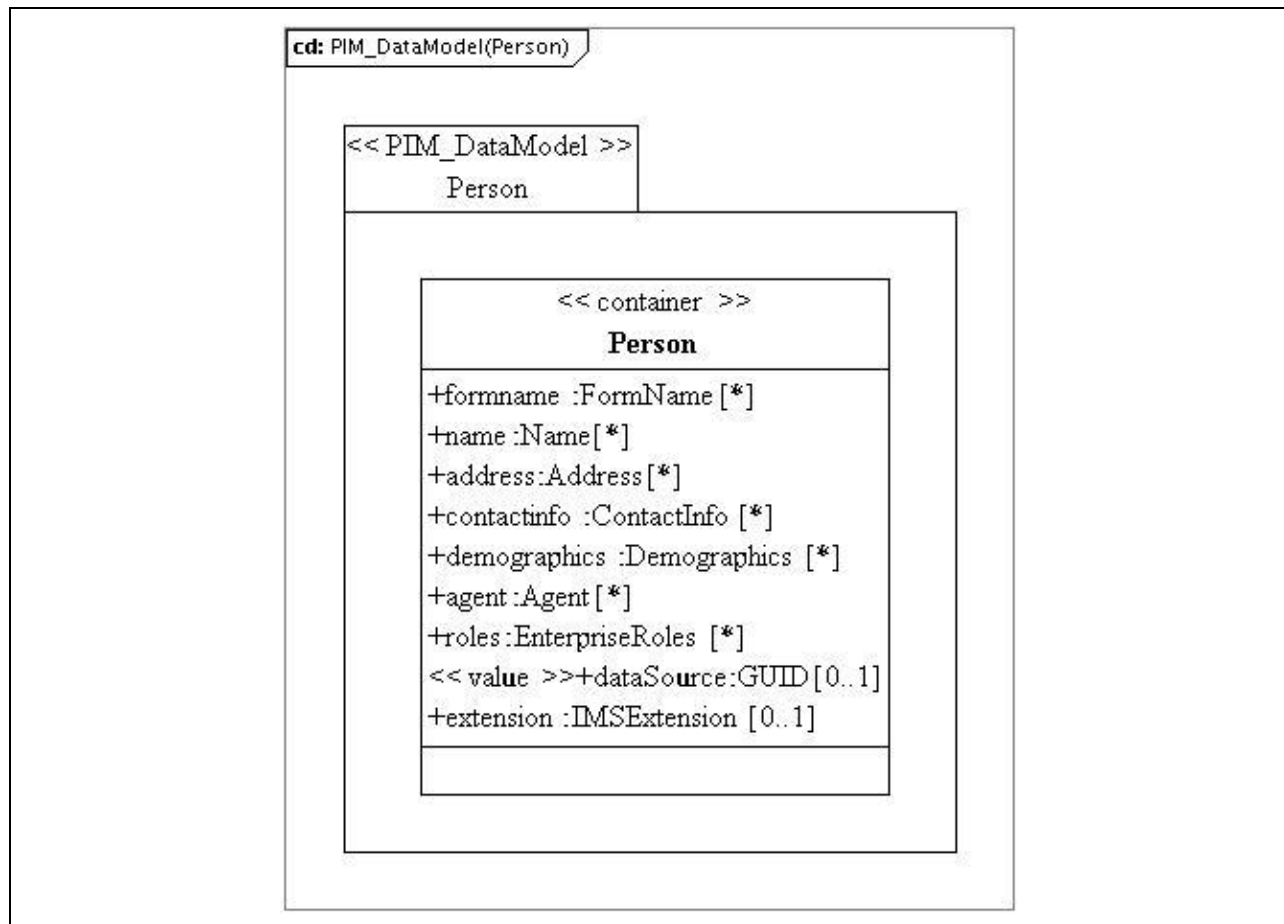
### 5.4.2 SourcedId Attribute Description

**Table 5.8 Description of the 'sourcedId' attribute for the SourcedGUID class.**

Descriptor	Definition
Attribute name	sourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	The sourcedId for the object. This should be a GUID.

## 5.5 Person Class Description

The PIM for the Person data model is shown in Figure 5.2.



**Figure 5.2 Person class diagram.**



**Table 5.9 Description of the ‘Person’ class.**

Descriptor	Definition
Class name	Person
Class type	container
Multiplicity	0..unbounded, unordered
Parents	PersonRecord
Children	[ formname, name, address, contactinfo, demographics, agent, roles, dataSource, extension ], ordered
Description	This is the container for all of the information about a single person.

### 5.5.1 Formname Attribute Description

**Table 5.10 Description of the ‘formname’ attribute for the Person class.**

Descriptor	Definition
Attribute name	formname
Data type	FormName
Value space	container
Multiplicity	0..unbounded, unordered
Description	The full names formatted in the way that the Person prefers to be addressed.

### 5.5.2 Name Attribute Description

**Table 5.11 Description of the ‘name’ attribute for the Person class.**

Descriptor	Definition
Attribute name	name
Data type	Name
Value space	container
Multiplicity	0..unbounded, unordered
Description	The names of the Person. This is an alternative view, to the ‘formname’ of the name. The name consists of one or parts e.g., first name, last name, etc.

### 5.5.3 Address Attribute Description

**Table 5.12 Description of the ‘address’ attribute for the Person class.**

Descriptor	Definition
Attribute name	address
Data type	Address
Value space	container
Multiplicity	0..unbounded, unordered
Description	The detailed address(es) of the person.

### 5.5.4 ContactInfo Attribute Description

**Table 5.13 Description of the ‘contactinfo’ attribute for the Person class.**

Descriptor	Definition
Attribute name	contactinfo
Data type	ContactInfo
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of electronic contact information e.g., telephone number(s), email address(es), etc.

### 5.5.5 Demographics Attribute Description

**Table 5.14 Description of the ‘demographics’ attribute for the Person class.**

Descriptor	Definition
Attribute name	demographics
Data type	Demographics
Value space	container
Multiplicity	0..unbounded, unordered
Description	The mechanisms by which the person can be recognized for learning.

### 5.5.6 Agent Attribute Description

**Table 5.15 Description of the ‘agent’ attribute for the Person class.**

Descriptor	Definition
Attribute name	agent
Data type	Agent
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of representatives e.g., guardian, etc. that can act on behalf of the person.

### 5.5.7 Roles Attribute Description

**Table 5.16 Description of the ‘roles’ attribute for the Person class.**

Descriptor	Definition
Attribute name	roles
Data type	EnterpriseRoles
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of roles that the person has within the Learning Information System.

### 5.5.8 DataSource Attribute Description

**Table 5.17 Description of the ‘dataSource’ attribute for the Person class.**

Descriptor	Definition
Attribute name	dataSource
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	An identifier of the source system of the object.

### 5.5.9 Extension Attribute Description

**Table 5.18 Description of the 'extension' attribute for the Person class.**

Descriptor	Definition
Attribute name	extension
Data type	IMSExtension
Value space	container
Multiplicity	0..1
Description	The extension mechanism for the Person data model.

## 5.6 FormName Class Description

The PIM for the FormName data model is shown in Figure 5.3.

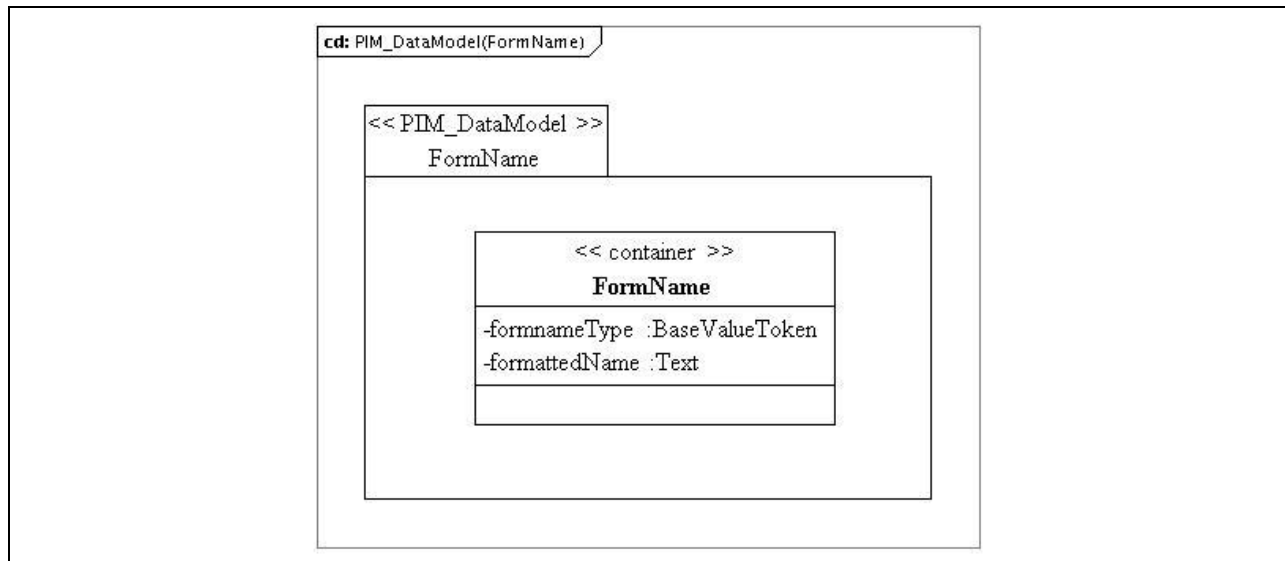


Figure 5.3 FormName class diagram.

Table 5.19 Description of the 'FormName' class.

Descriptor	Definition
Class name	FormName
Class type	container
Parents	Person and PersonCore
Children	[ formnameType, formattedName ], ordered
Description	The formatted name of the 'Person'.

### 5.6.1 FormnameType Attribute Description

**Table 5.20 Description of the ‘formnameType’ attribute for the FormName class.**

Descriptor	Definition
Attribute name	formnameType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Table B1.1.
Multiplicity	1
Description	<p>The data that is used to describe the type of the formatted name. Each type of formatted name is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

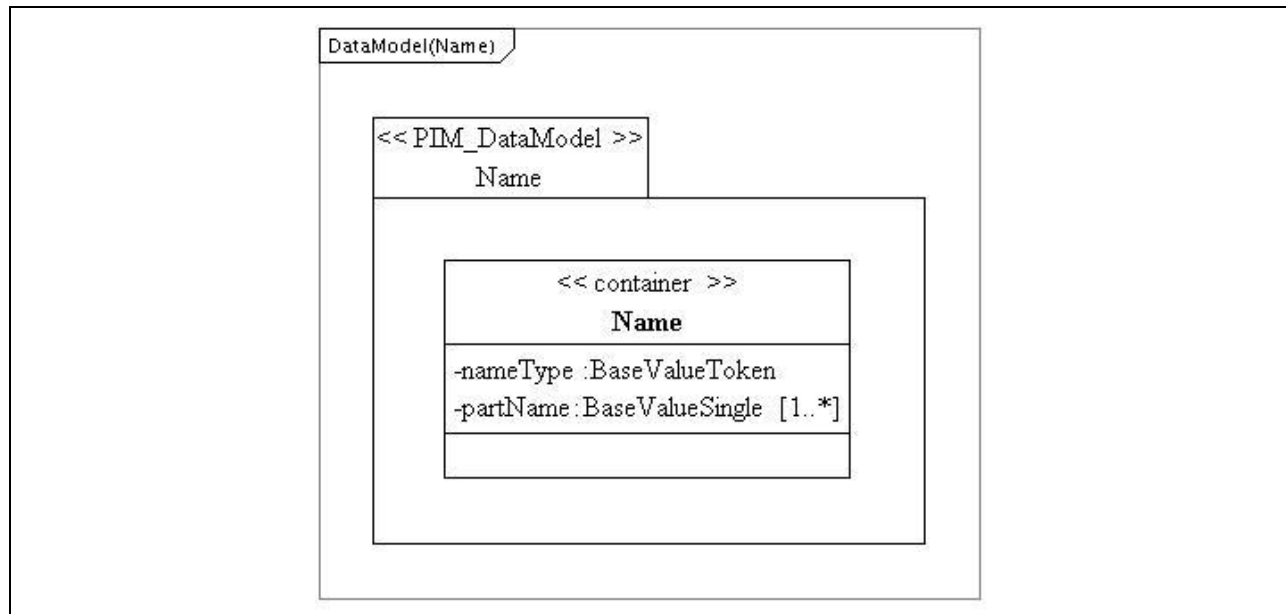
### 5.6.2 FormattedName Attribute Description

**Table 5.21 Description of the ‘formattedName’ attribute for the FormName class.**

Descriptor	Definition
Attribute name	formattedName
Data type	Text
Value space	Language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	Full name formatted in the way that the Person prefers to be addressed. There is no defined format for the name.

## 5.7 Name Class Description

The PIM for the Name data model is shown in Figure 5.4.



**Figure 5.4 Name class diagram.**

**Table 5.22 Description of the 'Name' class.**

Descriptor	Definition
Class name	Name
Class type	container
Multiplicity	0..unbounded, unordered
Parents	Person
Children	[ nameType, partName ], ordered
Description	The detailed name of the party. The name is supplied in its individual constituent parts.

### 5.7.1 NameType Attribute Description

**Table 5.23 Description of the 'nameType' attribute for the Name class.**

Descriptor	Definition
Attribute name	nameType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Table B1.2.
Multiplicity	1
Description	<p>The data that is used to describe the type of the name. Each type of name is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.7.2 PartName Attribute Description

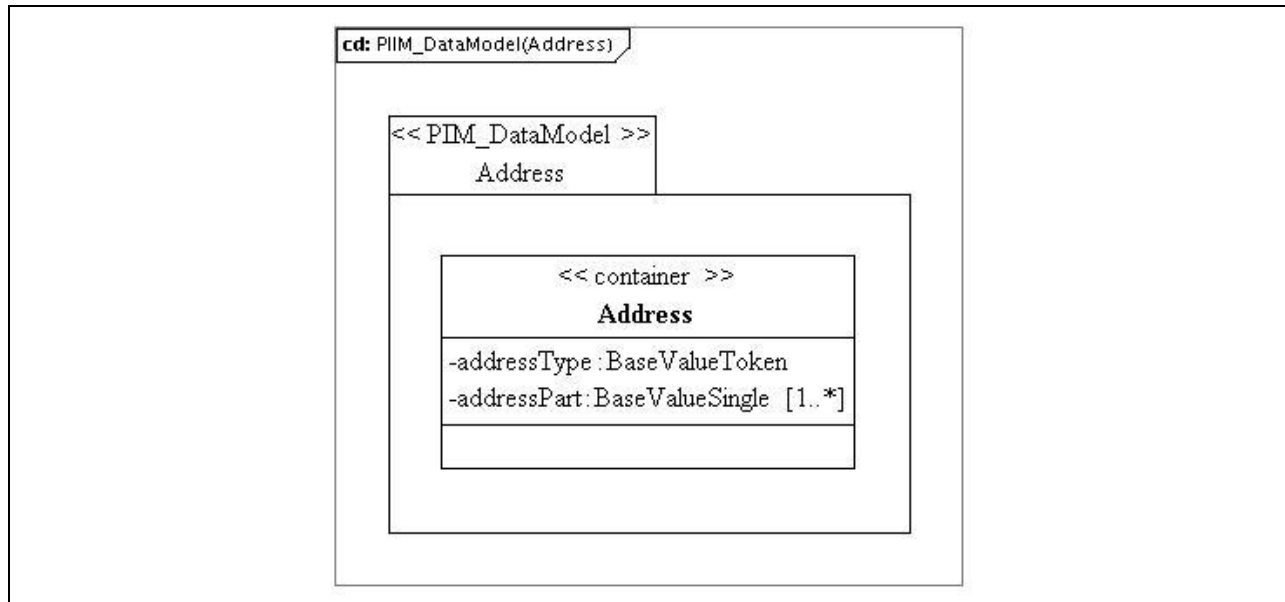
**Table 5.24 Description of the 'partName' attribute for the Name class.**

Descriptor	Definition
Attribute name	partName
Data type	BaseValueSingle
Value space	<p>The type of partName is vocabulary-based. The core vocabulary is given in Table B1.3.</p> <p>The value of the 'partName' is a language dependent String [1-255 characters]. The default language is 'en-US'.</p>
Multiplicity	1..unbounded, unordered (implementations must support at least 5 partName instances)
Description	<p>The part of the name being defined e.g., 'first', 'last', etc. A separate entry is used for each part of the name.</p> <p>Each type of name-part is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>



## 5.8 Address Class Description

The PIM for the Address data model is shown in Figure 5.5.



**Figure 5.5 Address class diagram.**

**Table 5.25 Description of the 'Address' class.**

Descriptor	Definition
Class name	Address
Class type	Container
Multiplicity	0..unbounded, unordered
Parents	Person
Children	[ addressType, addressPart ]
Description	The detailed address of the individual or organization.

### 5.8.1 AddressType Attribute Description

**Table 5.26 Description of the ‘addressType’ attribute for the Address class.**

Descriptor	Definition
Attribute name	addressType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.4.
Multiplicity	1
Description	<p>The data that is used to describe the type of the address. Each type of address is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.8.2 AddressPart Attribute Description

**Table 5.27 Description of the ‘addressPart’ attribute for the Address class.**

Descriptor	Definition
Attribute name	addressPart
Data type	BaseValueSingle
Value space	<p>The type of addressPart is Vocabulary-based. The core vocabulary is given in Appendix B1.5.</p> <p>The value of each ‘addressPart’ is a language dependent String [1-255 characters]. The default language is ‘en-US’.</p>
Multiplicity	1..unbounded, unordered (implementations must support at least 5 addressParts)
Description	<p>The address part. Each address part is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

## 5.9 ContactInfo Class Description

The PIM for the ContactInfo data model is shown in Figure 5.6.

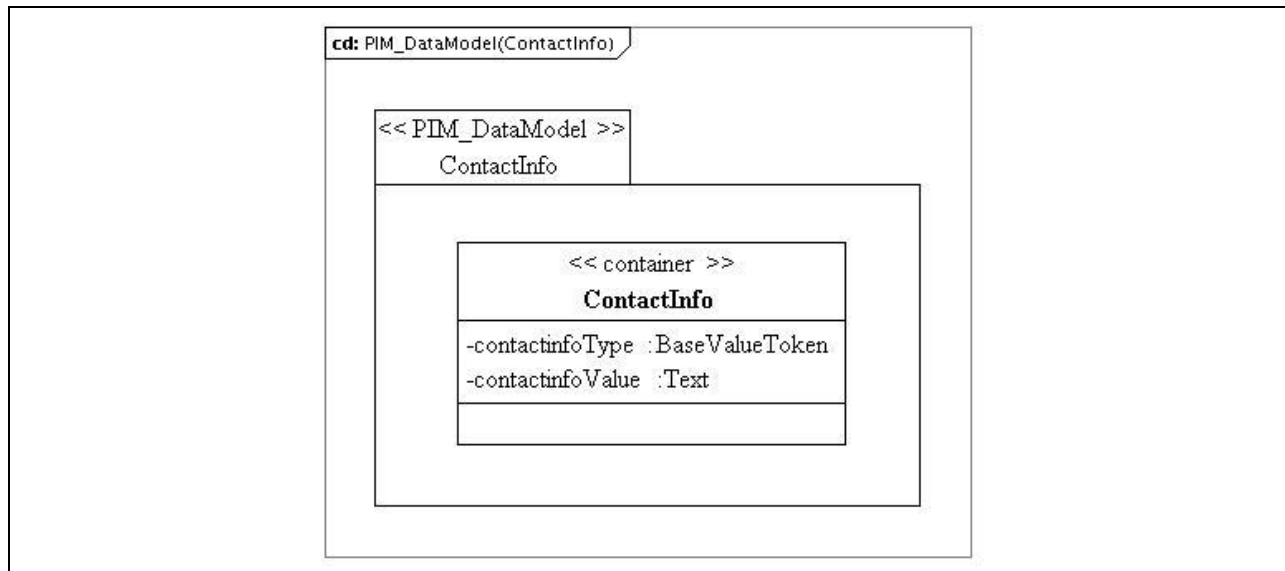


Figure 5.6 ContactInfo class diagram.

Table 5.28 Description of the 'ContactInfo' class.

Descriptor	Definition
Class name	ContactInfo
Class type	Container
Multiplicity	0..unbounded, unordered
Parents	Person
Children	[ contactinfoType, contactinfoValue ]
Description	The detailed contact information of the individual or organization

**Table 5.29 Description of the ‘contactinfoType’ attribute for the ContactInfo class.**

Descriptor	Definition
Attribute name	contactinfoType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.6.
Multiplicity	1
Description	<p>The data that is used to describe the type of the contact information. Each type of contact information is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p> <p>The vocabulary covers the type of contact information e.g., telephone, email, web address; the locality of the information e.g., telephone-home, email-work; and order of significance e.g., telephone-work-primary, telephone-work-secondary.</p>

**Table 5.30 Description of the ‘contactinfoValue’ attribute for the ContactInfo class.**

Descriptor	Definition
Attribute name	contactinfoValue
Data type	Text
Value space	A language dependent NormalizedString [1-127 characters]. Default language is ‘en-US’.
Multiplicity	1
Description	The contact information itself. The format for this information will depend upon the type of the contact information e.g., telephone number, email address, etc.

## 5.10 Demographics Class Description

The PIM for the Demographics data model is shown in Figure 5.7.

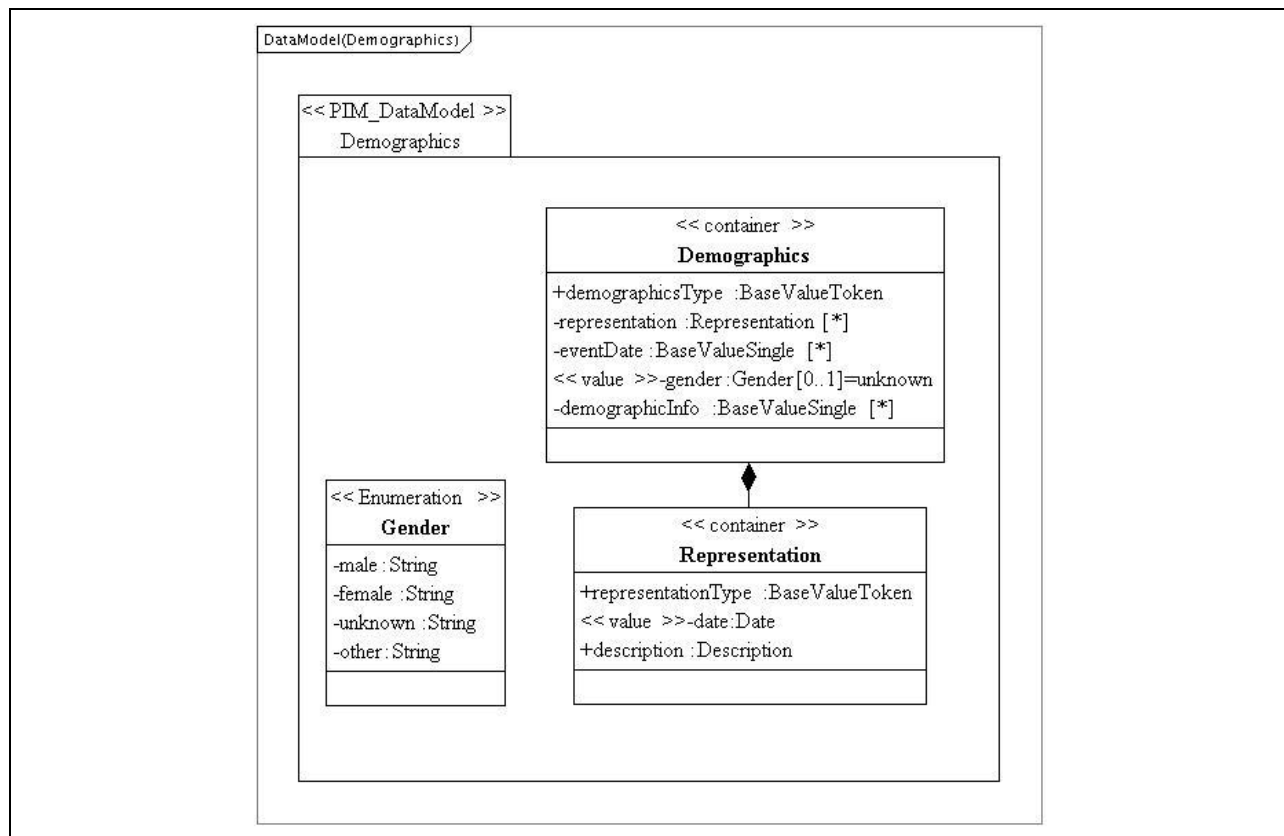


Figure 5.7 Demographics class diagram.

Table 5.31 Description of the ‘Demographics’ class.

Descriptor	Definition
Class name	Demographics
Class type	Container
Multiplicity	0..unbounded, unordered
Parents	Person
Children	[ demographicsType, representation, eventDate, gender, demographicsInfo ], ordered
Description	The mechanisms by which the individual can be recognized for learning.

### 5.10.1 DemographicsType Attribute Description

**Table 5.32 Description of the ‘demographicsType’ attribute for the Demographics class.**

Descriptor	Definition
Attribute name	demographicsType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.7.
Multiplicity	1
Description	<p>The data that is used to describe the type of the demographics. Each type of demographic is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

**Table 5.33 Description of the ‘Representation’ attribute for the Demographics class.**

Descriptor	Definition
Attribute name	Representation
Data type	Representation
Value space	Container
Multiplicity	0..unbounded, unordered (an implementation should support at least one instance)
Description	Representative information of the learner e.g., photograph.

### 5.10.2 Dates Attribute Description

**Table 5.34 Description of the ‘eventDate’ attribute for the Demographics class.**

Descriptor	Definition
Attribute name	eventDate
Data type	BaseValueToken
Value space	The name is vocabulary based with a core vocabulary given in Appendix B1.9. The format for the date value is ISO8601 i.e., ‘YYYY-MM-DD’.
Multiplicity	0..unbounded, unordered
Description	Recorded dates appropriate to the demographics data e.g., date of birth, etc.

### 5.10.3 Gender Attribute Description

**Table 5.35 Description of the ‘gender’ attribute for the Demographics class.**

Descriptor	Definition
Attribute name	Gender
Data type	Enumerated
Value space	{ male, female, unknown, other }. The default value is ‘unknown’.
Multiplicity	0..1
Description	The gender of the learner.

### 5.10.4 DemographicInfo Attribute Description

**Table 5.36 Description of the ‘demographicInfo’ attribute for the Demographics class.**

Descriptor	Definition
Attribute name	demographicInfo
Data type	BaseValueSingle
Value space	<p>The type of demographicsInfo is vocabulary-based. The core vocabulary is given in Appendix B1.8.</p> <p>The value of the ‘demographicsInfo’ is a language dependent String [1-255 characters]. The default language is ‘en-US’.</p>
Multiplicity	0..unbounded, unordered (an implementation should support at least 1 instance)
Description	<p>The type of the demographics being defined e.g., ‘PlaceofBirth’, ‘MaritalStatus’, etc.</p> <p>The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

## 5.11 Representation Class Description

**Table 5.37 Description of the ‘Representation’ class.**

Descriptor	Definition
Class name	Representation
Class type	Container
Multiplicity	0..unbounded, unordered
Parents	Demographics
Children	[ representationType, date, description ]
Description	Representative information of the learner



### 5.11.1 RepresentationType Attribute Description

**Table 5.38 Description of the ‘representationType’ attribute of the Representation class.**

Descriptor	Definition
Attribute name	representationType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.10.
Multiplicity	1
Description	<p>The data that is used to describe the type of the representation. Each type of representation is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

**Table 5.39 Description of the ‘date’ attribute of the Representation class.**

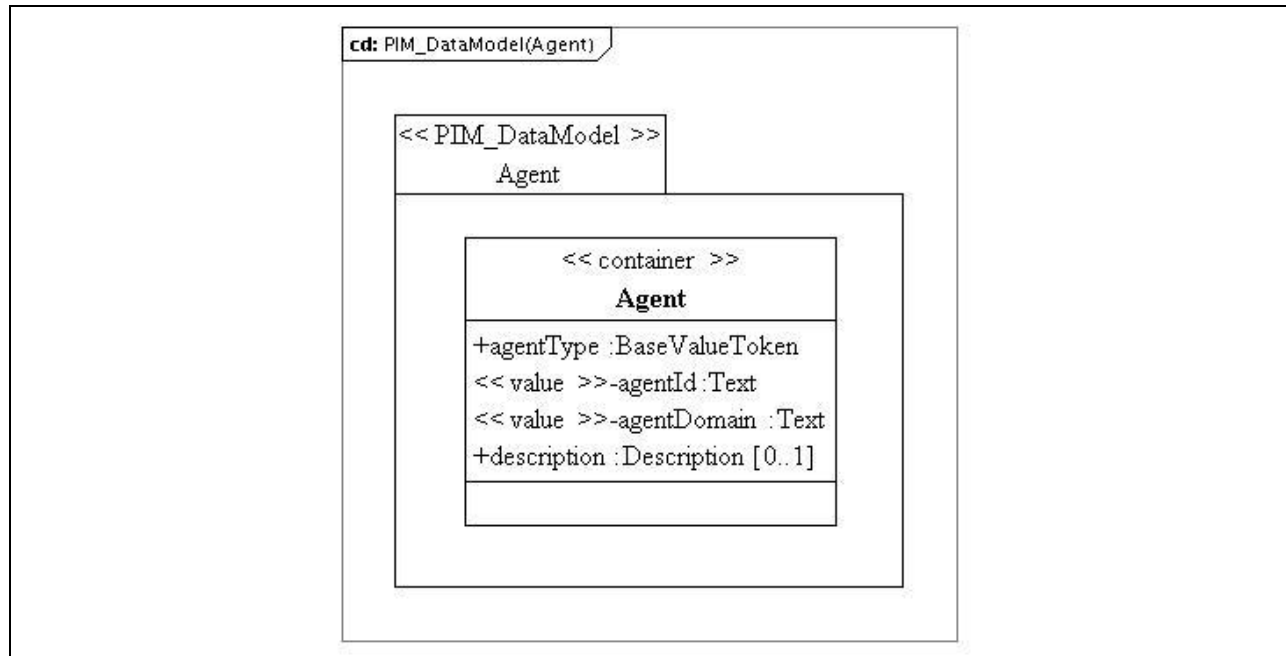
Descriptor	Definition
Attribute name	Date
Data type	Date
Value space	ISO 8601 format (YYYY-MM-DD).
Multiplicity	1
Description	Recorded dates appropriate to the representation data.

**Table 5.40 Description of the ‘Description’ attribute of the Representation class.**

Descriptor	Definition
Attribute name	Description
Data type	Description
Value space	Container
Multiplicity	1
Description	The representation of the learner and a brief description of its usage.

## 5.12 Agent Class Description

The PIM for the Agent data model is shown in Figure 5.8.



**Figure 5.8 Agent class diagram.**

**Table 5.41 Description of the ‘Agent’ class.**

Descriptor	Definition
Class name	Agent
Class type	Container
Multiplicity	0..unbounded, unordered
Parents	Person
Children	[ agentType, agentId, agentDomain, description ], ordered
Description	The information of the ‘agents’ who can act on behalf of the learner.

### 5.12.1 AgentType Attribute Description

**Table 5.42 Description of the ‘agentType’ attribute for the Agent class.**

Descriptor	Definition
Attribute name	agentType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.11.
Multiplicity	1
Description	<p>The data that is used to describe the type of the agent. Each type of agent is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.12.2 AgentId Attribute Description

**Table 5.43 Description of the ‘agentId’ attribute for the Agent class.**

Descriptor	Definition
Attribute name	agentId
Data type	Text
Value space	A language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The identifier assigned to the agent.

### 5.12.3 Agentdomain Attribute Description

**Table 5.44** Description of the ‘agentDomain’ attribute for the Agent class.

Descriptor	Definition
Attribute name	agentDomain
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The role of the agent e.g., legal, financial etc.

### 5.12.4 Description Attribute Description

**Table 5.45** Description of the ‘Description’ attribute for the Agent class.

Descriptor	Definition
Attribute name	description
Data type	Description
Value space	container
Multiplicity	0..1
Description	Description of the role of the agent.

### 5.13 EnterpriseRoles Class Description

The PIM for the EnterpriseRoles data model is shown in Figure 5.9.

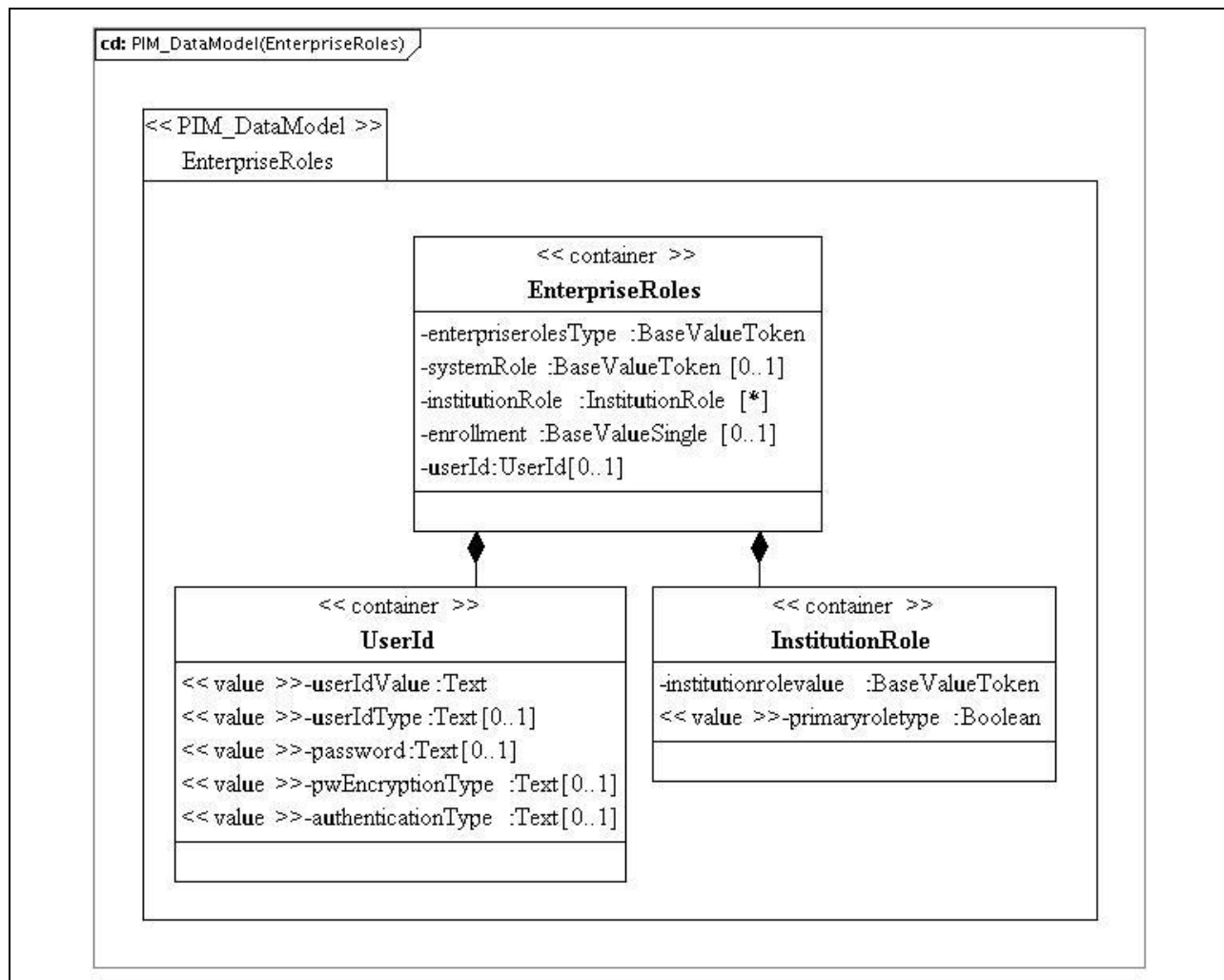


Figure 5.9 EnterpriseRoles class diagram.

Table 5.46 Description of the 'EnterpriseRoles' class.

Descriptor	Definition
Class name	EnterpriseRoles
Class type	Container
Multiplicity	1..unbounded, unordered
Parents	Person
Children	[ enterpriserolesType, systemRole, institutionRole, enrollment, userId, ], ordered
Description	The roles assigned to the person within the Enterprise system.

### 5.13.1 EnterpriseRolesType Attribute Description

**Table 5.47 Description of the ‘enterpriserolesType’ attribute for the EnterpriseRoles class.**

Descriptor	Definition
Attribute name	enterpriserolesType
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.12.
Multiplicity	1
Description	<p>The data that is used to describe the type of the enterprise role. Each type of roles is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.13.2 SystemRole Attribute Description

**Table 5.48 Description of the ‘systemRole’ attribute for the EnterpriseRoles class.**

Descriptor	Definition
Attribute name	systemRole
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.13.
Multiplicity	0..1
Description	<p>The data that is used to describe the system role. Each role is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.13.3 InstitutionRole Attribute Description

**Table 5.49 Description of the ‘institutionRole’ attribute for the EnterpriseRoles class.**

Descriptor	Definition
Attribute name	institutionRole
Data type	InstitutionRole
Value space	container
Multiplicity	0..*, unordered (an implementation should support at least 2 instances).
Description	The role of the Person within the institution

### 5.13.4 Enrollment Attribute Description

**Table 5.50 Description of the ‘enrollment’ attribute for the EnterpriseRoles class.**

Descriptor	Definition
Attribute name	enrollment
Data type	BaseValueSingle
Value space	<p>The type of enrollment is vocabulary-based. The core vocabulary is given in Appendix B1.15.</p> <p>The value of the ‘enrollment’ is a language dependent String [1-255 characters]. The default language is ‘en-US’.</p>
Multiplicity	0..unbounded, unordered (an implementation should support at least 1 instance)
Description	<p>The type of the enrollment being defined e.g., ‘AcedemicDegree’, ‘AcademicMajor’, etc.</p> <p>The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.13.5 UserId Attribute Description

**Table 5.51 Description of the ‘userId’ attribute for the EnterpriseRoles class.**

Descriptor	Definition
Attribute name	userId
Data type	UserId
Value space	n/a
Multiplicity	0..1
Description	The person’s user identifier to access the learning management environment.

## 5.14 UserId Class Description

**Table 5.52 Description of the ‘UserId’ class.**

Descriptor	Definition
Class name	UserId
Class type	container
Multiplicity	0..1
Parents	EnterpriseRoles and PersonCore
Children	[ Text ], ordered
Description	The container for the information that defines the access control for a user to the learning environment.

### 5.14.1 UserIdValue Attribute Description

**Table 5.53 Description of the ‘userIdValue’ attribute for the UserId class.**

Descriptor	Definition
Attribute name	userId
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The actual user identifier for access to the learning environment.



### 5.14.2 UserIdType Attribute Description

**Table 5.54 Description of the ‘userIdType’ attribute for the UserId class.**

Descriptor	Definition
Attribute name	userIdType
Data type	Text
Value space	A language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	The type of user identification. Some examples are ‘NationalId’, ‘InstitutionId’, etc.

### 5.14.3 Password Attribute Description

**Table 5.55 Description of the ‘password’ attribute for the UserId class.**

Descriptor	Definition
Attribute name	password
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	The password that is assigned to the user for accessing a system

### 5.14.4 PwEncryption Attribute Description

**Table 5.56 Description of the ‘pwEncryption’ attribute for the UserId class.**

Descriptor	Definition
Attribute name	pwEncryption
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	The type of encryption that has been applied to the password. Examples include ‘PKC’, ‘MD5’, ‘SHA-1’, ‘SHA-256’, etc.

### 5.14.5 AuthenticationType Attribute Description

**Table 5.57 Description of the ‘authenticationType’ attribute for the UserId class.**

Descriptor	Definition
Attribute name	authenticationType
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	The type of authentication that is to be applied to control access to the system. Example values include: SSO, LDAP, ActiveDirectory, Kerberos5, etc.

## 5.15 InstitutionRole Class Description

**Table 5.58 Description of the ‘InstitutionRole’ class.**

Descriptor	Definition
Class name	InstitutionRole
Class type	container
Multiplicity	0..unbounded, unordered (an implementation should support at least 2 instances).
Parents	EnterpriseRoles
Children	[ institutionrolevalue, primaryroletype ]
Description	The role of the Person within the institution. Each separate occurrence is used to define each role.

### 5.15.1 Institutionrolevalue Attribute Description

**Table 5.59 Description of the ‘institutionrolevalue’ attribute for the InstitutionRole class.**

Descriptor	Definition
Attribute name	institutionrolevalue
Data type	BaseValueToken
Value space	Vocabulary-based. The core vocabulary is given in Appendix B1.14.
Multiplicity	1
Description	<p>The role of the Person within the institution. Each role is uniquely identified and its value is based upon the corresponding vocabulary. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

### 5.15.2 Primaryroletype Attribute Description

**Table 5.60 Description of the ‘primaryroletype’ attribute for the InstitutionRole class.**

Descriptor	Definition
Attribute name	primaryroletype
Data type	Boolean
Value space	{ True = primary role; False = not primary role. }
Multiplicity	1
Description	Identifies if the associated role is the primary one for the Person in the institution. The primary role is denoted by a value of ‘true’.

## 5.16 Common Classes Descriptions

The PIM for the Common classes data model is shown in Figure 5.10.

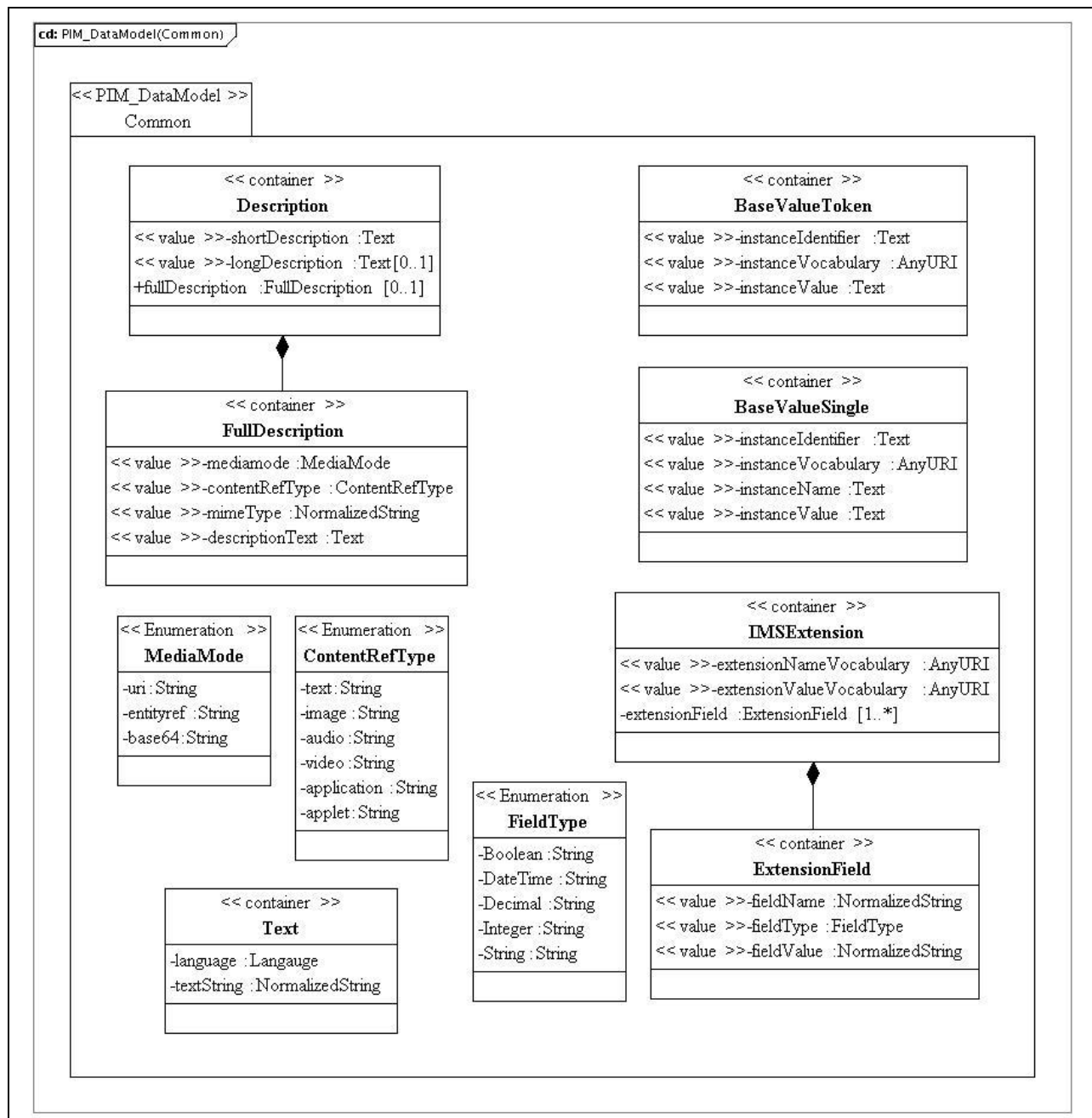


Figure 5.10 Common classes diagram.

### 5.16.1 BaseValueToken Class Description

**Table 5.61 Description of the ‘BaseValueToken’ class.**

Descriptor	Definition
Class name	BaseValueToken
Class type	container
Children	[ instanceIdentifier, instanceVocabulary, instanceValue ]
Description	The value from a vocabulary. This is used to provide, and uniquely identify, a selection from an enumerated list within a vocabulary.

**Table 5.62 Description of the ‘instanceIdentifier’ attribute for the BaseValueToken class.**

Descriptor	Definition
Attribute name	instanceIdentifier
Data type	Text
Value space	A language dependent String [1-4095 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The identifier of the object. This is used to provide a unique reference point for each component within the object.

**Table 5.63 Description of the ‘instanceVocabulary’ attribute for the BaseValueToken class.**

Descriptor	Definition
Attribute name	instanceVocabulary
Data type	AnyURI
Value space	NormalizedString [1-4095 characters].
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldName values. If this is a reference to a VDEX file it is the ‘vocabIdentifier’ of the vocabulary.

**Table 5.64 Description of the ‘instanceValue’ attribute for the BaseValueToken class.**

Descriptor	Definition
Attribute name	instanceValue
Data type	Text
Value space	A language dependent String [1-255 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The selected value from the vocabulary. This should be an entry in the vocabulary identified in the ‘instanceVocabulary’ entry.

### 5.16.2 BaseValueSingle Class Description

**Table 5.65 Description of the ‘BaseValueSingle’ class.**

Descriptor	Definition
Class name	BaseValueSingle
Class type	container
Children	[ instanceIdentifier, instanceVocabulary, instanceName, instanceValue ] , ordered
Description	The value for a name/value tuple with the name selected from a vocabulary.

**Table 5.66 Description of the ‘instanceIdentifier’ attribute for the BaseValueSingle class.**

Descriptor	Definition
Attribute name	instanceIdentifier
Data type	Text
Value space	A language dependent String [1-4095 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The identifier of the object. This is used to provide a unique reference point for each component within the object.

**Table 5.67 Description of the ‘instanceVocabulary’ attribute for the BaseValueSingle class.**

Descriptor	Definition
Attribute name	instanceVocabulary
Data type	AnyURI
Value space	NormalizedString [1-4095 characters].
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldName values. If this is a reference to a VDEX file it is the ‘vocabIdentifier’ of the vocabulary.

**Table 5.68 Description of the ‘instanceName’ attribute for the BaseValueSingle class.**

Descriptor	Definition
Attribute name	instanceName
Data type	Text
Value space	A language dependent Normalized String [1-4095 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The name of the name/value tuple. The name components must be contained within the given vocabulary.

**Table 5.69 Description of the ‘instanceValue’ attribute for the BaseValueSingle class.**

Descriptor	Definition
Attribute name	instanceValue
Data type	Text
Value space	A language dependent NormalizedString [1-1027 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The value of the name/value tuple.

### 5.16.3 Description Class Description

**Table 5.70 Description of the ‘Description’ class.**

Descriptor	Definition
Class name	Description
Class type	container
Children	[ shortDescription, longDescription, fullDescription ]
Description	The container for descriptive material about the associated object. The description can take the form of text, image, video, audio, etc.

**Table 5.71 Description of the ‘shortDescription’ attribute for the Description class.**

Descriptor	Definition
Attribute name	shortDescription
Data type	Text
Value space	A language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	A short textual description.

**Table 5.72 Description of the ‘longDescription’ attribute for the Description class.**

Descriptor	Definition
Attribute name	longDescription
Data type	Text
Value space	A language dependent String [1-2095 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	A long textual description.



**Table 5.73 Description of the ‘fullDescription’ attribute for the Description class.**

Descriptor	Definition
Attribute name	fullDescription
Data type	FullDescription
Value space	container
Multiplicity	0..1
Description	The full description consists of the required material types.

#### 5.16.4 FullDescription Class Description

**Table 5.74 Description of the ‘FullDescription’ class.**

Descriptor	Definition
Class name	FullDescription
Class type	container
Parents	Description
Children	[ mediamode, contentRefType, mimeType, descriptionText ]
Description	A full description of the activity, etc. using text, images, etc.

**Table 5.75 Description of the ‘mediaMode’ attribute for the FullDescription class.**

Descriptor	Definition
Attribute name	mediaMode
Data type	Enumerated
Value space	{ uri, entityref, base64 }
Multiplicity	1
Description	The reference form to the material. ‘uri’ identifies the description as an external file with a URI; ‘entityref’ means the description is linked as a entityref within the XML file; and ‘base64’ means the description is base64 encoded within the XML file.

**Table 5.76 Description of the ‘contentRefType’ attribute for the FullDescription class.**

Descriptor	Definition
Attribute name	contentRefType
Data type	Enumerated
Value space	{ text, image, audio, video, application, applet }
Multiplicity	1
Description	The type of the material.

**Table 5.77 Description of the ‘mimeType’ attribute for the FullDescription class.**

Descriptor	Definition
Attribute name	mimeType
Data type	NormalizedString
Value space	Normalized String [1..63] characters.
Multiplicity	1
Description	The mime-type for the material.

**Table 5.78 Description of the ‘descriptionText’ attribute for the FullDescription class.**

Descriptor	Definition
Attribute name	descriptionText
Data type	Text
Value space	A language dependent String [1-1027 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	A textual description of the material.

### 5.16.5 Text Class Description

**Table 5.79 Description of the ‘Text’ class.**

Descriptor	Definition
Class name	Text
Class type	container
Children	[ language, textString ]
Description	Text to be stored. This is a language/string tuple.

**Table 5.80 Description of the ‘language’ attribute for the Text class.**

Descriptor	Definition
Attribute name	language
Data type	Enumerated.
Value space	RFC4646 language code-country code combination. The default value is: ‘en-US’.
Multiplicity	1
Description	The language for the associated text string.

**Table 5.81 Description of the ‘textString’ attribute for the Text class.**

Descriptor	Definition
Attribute name	textString
Data type	Normalized String
Value space	Normalized String [1-4095 characters].
Multiplicity	1
Description	The container for the string.

### 5.16.6 IMSEExtension Class Description

The PIM for the Extension class is shown in Figure 5.10.

**Table 5.82 Description of the IMSEExtension class.**

Descriptor	Definition
Class name	IMSEExtension
Class type	container
Children	[ extensionNameVocabulary, extensionTypeVocabulary, extensionField ], ordered
Description	The container for the extension of the Membership data model.

**Table 5.83 Description of the ‘extensionNameVocabulary’ attribute for the IMSEExtension class.**

Descriptor	Definition
Attribute name	ExtensionNameVocabulary
Data type	AnyURI
Value space	Normalized String [1-4095 characters].
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldName values. If this is a reference to a VDEX file it is the ‘vocabIdentifier’ of the vocabulary.

**Table 5.84 Description of the ‘extensionTypeVocabulary’ attribute for the IMSEExtension class.**

Descriptor	Definition
Attribute name	extensionTypeVocabulary
Data type	AnyURI
Value space	Normalized String [1-4095 characters].
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldType values. If this is a reference to a VDEX file it is the ‘vocabIdentifier’ of the vocabulary.

**Table 5.85 Description of the ‘extensionField’ attribute for the IMSExtension class.**

Descriptor	Definition
Attribute name	extensionField
Data type	ExtensionField
Value space	container
Multiplicity	1..unbounded, unordered
Description	The container for the triples that are used to define each extension data element.

### 5.16.7 ExtensionField Class Description

The PIM for the Metadata class is shown in Figure 5.10.

**Table 5.86 Description of the ExtensionField class.**

Descriptor	Definition
Class name	ExtensionField
Class type	container
Children	None.
Description	The container for each triple that describes an extension field. Each triple consists of field name, field type and field value.

**Table 5.87 Description of the ‘fieldName’ attribute for the ExtensionField class.**

Descriptor	Definition
Attribute name	fieldName
Data type	NormalizedString
Value space	A language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The container for the name of the extension field. This is used to identify the full triple. This must be a value from the associated vocabulary.

**Table 5.88 Description of the ‘fieldType’ attribute for the ExtensionField class.**

Descriptor	Definition
Attribute name	fieldType
Data type	Enumerated vocabulary.
Value space	Vocabulary-based. The core vocabulary is given in Appendix B.16.
Multiplicity	1
Description	<p>This defines the data-type for the extension triple. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model.</p> <p>The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.</p>

**Table 5.89 Description of the ‘fieldValue’ attribute for the ExtensionField class.**

Descriptor	Definition
Attribute name	fieldValue
Data type	NormalizedString
Value space	A language dependent String [1-1023 characters]. The default language is ‘en-US’.
Multiplicity	1
Description	The container for the data value itself. This is stored as a string but should be interpreted as per the data-type defined in the ‘fieldType’ part of the triple.

## 6 Extending and Profiling the Service

### 6.1 Proprietary Extensions

Proprietary extensions of the service are based upon two approaches:

- a) The extension of the data models being manipulated by the current set of operations;
- b) The inclusion of new operations to support new proprietary functionality.

It is NOT permitted to change the behavior of the current set of operations. Such changes MUST be supported by the creation of new operations.

#### 6.1.1 Proprietary Operations

The definition of new operations should follow the same format as adopted herein. The new operations should be defined using a new interface type. Every operation must result in the return of a status code that describes the final state of the request on the target end system.

An example of creating such an extension is given in the accompanying Best Practices document [LIS, 11c].

#### 6.1.2 Proprietary Data Elements

Extensions to the data model are only permitted where the *IMSExtension* class is available. Within the Person data model only the 'Person' class can be extended. The extension takes the form of a Name/Type/Value triple. Many extension fields can be added but hierarchical structures must be emulated using the appropriate delimited notation in the 'Name' field. This triple consists of:

- Name – the name assigned to the extension field (this is a string that can support any naming convention);
- Type – the data-type that is to be used for the value (this is used for interpreting the associated value);
- Value – the data value for the extension (the value is supplied as a string).

### 6.2 Profiling the Service

This Service can be profiled. In general, Profiling is used to:

- a) Refine which Interfaces are used and which operations are supported for each Interface;
- b) Refine the data models (see the IMS GLC Application Profiling guidelines for more details on how data models can be profiled [APG, 05a][APG, 05b]).

Valid Profiles must be restrictive i.e., optional features can be removed or constraints increased but new features must not be added. A Profile of this service is made by annotating the UML supplied with the documentation for the specification.

## Appendix A – Service Status Codes

The summary list of status codes that can be returned by the different operations through the StatusInfo object is given in Table A.1. The key to the entries is: ‘Y’ denotes the code may be returned by that operation. A blank entry means that the code cannot be returned by that operation.

**Table A.1 Status codes for the PersonManager interface operations.**

CodeMinor Status Code	create	createByProxy	delete	read <sup>1</sup>	update	replace	discover	changeIdentifier
‘fullsuccess’	Y	Y	Y	Y	Y	Y	Y	Y
‘createsuccess’						Y		
‘nosourcedids’				Y			Y	
‘idallocfail’		Y						
‘overflowfail’	Y	Y						
‘idallocinusefail’	Y							Y
‘invaliddata’	Y	Y		Y	Y	Y		
‘incompletedata’	Y	Y		Y	Y	Y		
‘partialdatastorage’	Y	Y		Y	Y	Y		
‘unknownobject’			Y	Y	Y			Y
‘deletefailure’			Y					
‘toomuchdata’				Y			Y	
‘targetreadfailure’				Y				
‘savepointerror’				Y				
‘savepointsyncerror’				Y				
‘unknownquery’							Y	
‘unknownvocabulary’	Y	Y		Y	Y	Y		
‘unknownmdvocabulary’	Y	Y			Y	Y		
‘unknownnextension’	Y	Y		Y	Y	Y		
‘targetisbusy’	Y	Y	Y	Y	Y	Y	Y	Y
‘unauthorizedrequest’	Y	Y	Y	Y	Y	Y	Y	Y
‘linkfailure’	Y	Y	Y	Y	Y	Y	Y	Y
‘unsupportedLIS’	Y	Y	Y	Y	Y	Y	Y	Y
‘unsupportedLISOperation’	Y	Y	Y	Y	Y	Y	Y	Y



Note:

1. ‘\*’ denotes used by the ‘readPerson’, ‘readPersonCore’, ‘readPersons’, ‘readAllPersonIds’, ‘readPersonsFromSavePoint’, and ‘readPersonIdsFromSavePoint’ operations.

There is a set of status codes that must be supported by each of the Person Management Service operations. These codes are described in Table A.2.

**Table A.2 Common status codes for the service operations.**

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unauthorizedrequest’	The source system is not authorized to make this request of the target. The reason for the refusal can be one of several causes.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=targetisbusy’	The target end-system received the request but is busy and cannot process the request. The request should be resubmitted.
‘CodeMajor=Failure’ ‘Severity=Error’ ‘CodeMinor=linkfailure’	There has been a failure in the end-to-end system communications mechanism and so the request has not been delivered.
‘CodeMajor=UnsupportedLIS’ ‘Severity=Status’ ‘CodeMinor=*’	This service in LIS is not supported by the target system. Every system that implements any part of the LIS specification <b>must</b> return this status code for a service component in LIS that is not supported.
‘CodeMajor=UnsupportedLISOperation’ ‘Severity=Status’ ‘CodeMinor=*’	This operation is not supported by the target system. Every system that implements any part of the LIS specification <b>must</b> return this status code for an operation that is not supported in a supported service.

## Appendix B – Set of Vocabularies

### B1 Set of Defined Vocabularies

The vocabularies listed in the following Tables are the default set maintained under the IMS GLC Vocabulary Registry [SDN11, 06]. It is the responsibility of an implementation to ensure that it is using the correct and latest versions of the vocabulary files. Changes to the default vocabularies are permitted; this results in the creation of a new vocabulary that should be registered with IMS GLC. As part of a profiling process entirely new vocabularies may be defined to replace the default set.

#### B1.1 FormnameType Vocabulary

The vocabulary for ‘formnameType’ is listed in Table B1.1.

**Table B1.1 Vocabulary for ‘formnameType’.**

Vocabulary Association	Description of the Vocabulary
Type of formatted name: ‘formnameType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"><li>• Alias;</li><li>• Contact;</li><li>• Former;</li><li>• Full;</li><li>• Maiden;</li><li>• Preferred.</li></ul>

Rule B.1-01: Alias – an alternative name as a human readable single string;

Rule B.1-02: Contact – the name to be used in all formal contact/correspondence as a human readable single string;

Rule B.1-03: Former – a former name as a human readable single string;

Rule B.1-04: Full – this is the full name as a human readable single string;

Rule B.1-05: Maiden – the full maiden name as a human readable single string;

Rule B.1-05: Preferred – the full name of preference as a human readable single string.

#### B1.2 NameType Vocabulary

The vocabulary for ‘formnameType’ is listed in Table B1.2.

**Table B1.2 Vocabulary for ‘nameType’.**

Vocabulary Association	Description of the Vocabulary
Type of name: ‘nameType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"><li>• Alias;</li><li>• Contact;</li><li>• Former;</li><li>• Full;</li><li>• Maiden;</li><li>• Preferred.</li></ul>

Rule B.2-01: Alias – an alternative name to be given as a set of composite strings;

Rule B.2-02: Contact – the name to be used in all formal contact/correspondence to be given as a set of composite strings;

Rule B.2-03: Former – a former name to be given as a set of composite strings;

Rule B.2-04: Full – this is the full name to be given as a set of composite strings;

Rule B.2-05: Maiden – the full maiden name to be given as a set of composite strings;

Rule B.2-05: Preferred – the full name of preference to be given as a set of composite strings.

### B1.3 PartName Vocabulary

The vocabulary for type of 'partName' is listed in Table B1.3.

**Table B1.3 Vocabulary for type of 'partName'.**

Vocabulary Association	Description of the Vocabulary
Type of part name: 'partName'	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Family;</li> <li>• First;</li> <li>• Given;</li> <li>• Initials;</li> <li>• Last;</li> <li>• Maternal;</li> <li>• Middle;</li> <li>• Nickname;</li> <li>• Particle;</li> <li>• Patrilineal;</li> <li>• Prefix;</li> <li>• Suffix;</li> <li>• Surname.</li> </ul>

Rule B.3-01: Family – the assigned family name (this is also as known as a type of surname);

Rule B.3-02: First – the assigned first name;

Rule B.3-03: Given – the assigned given name (also known as first name, forename or personal name);

Rule B.3-04: Initials – the set of initials assigned to the individual (using a comma separated list);

Rule B.3-05: Last – the assigned last name (also known as the family or surname);

Rule B.3-06: Maternal – the maternal surname i.e., the surname inherited from the mother's father;

Rule B.3-07: Middle – a middle name assigned to an individual (if more than one middle name is assigned then multiple middle name parts will be defined);

Rule B.3-08: Nickname – a nick name, in full, assigned to the individual;

Rule B.3-09: Particle – the particle of the name e.g., 'de', 'y', 'van', etc;

Rule B.3-09: Patrilineal – the surname inherited from the father;

Rule B.3-11: Prefix – a prefix to the name e.g., Miss, Mr, Dr, Reverend, etc. (if more than one prefix is assigned then multiple prefix name parts will be defined);

Rule B.3-12: Suffix – a suffix to the name e.g., III, BSc, PhD, MIEEE, etc. (if more than one suffix is assigned then multiple suffix name parts will be defined);

Rule B.3-13: Surname – the assigned surname (also known as the last or family name).

#### **B1.4 AddressType Vocabulary**

The vocabulary for the 'addressType' is listed in Table B1.4.

**Table B1.4 Vocabulary for the 'addressType'.**

Vocabulary Association	Description of the Vocabulary
Type of address: 'addressType'	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Billing_Primary;</li> <li>• Campus_Primary;</li> <li>• Home_Primary;</li> <li>• Mailing_Primary;</li> <li>• Permanent_Primary;</li> <li>• Private_Primary;</li> <li>• Temporary_Primary;</li> <li>• Work_Primary;</li> <li>• Billing_Secondary;</li> <li>• Campus_Secondary;</li> <li>• Home_Secondary;</li> <li>• Mailing_Secondary;</li> <li>• Permanent_Secondary</li> <li>• Private_Secondary</li> <li>• Temporary_Secondary;</li> <li>• Work_Secondary.</li> </ul>

Rule B.4-01: Billing\_Primary – the primary billing address;

Rule B.4-02: Campus\_Primary – the primary contact address for the individual on campus;

Rule B.4-03: Home\_Primary – the primary contact home address for the individual;

Rule B.4-04: Mailing\_Primary – the primary postal mailing address for the individual;

Rule B.4-05: Permanent\_Primary – the primary permanent address for the individual;

Rule B.4-06: Private\_Primary – the primary private contact address for the individual;

Rule B.4-07: Temporary\_Primary – the primary temporary contact address for the individual;

Rule B.4-08: Work\_Primary – the primary work contact address for the individual;

Rule B.4-09: Billing\_Secondary – the secondary billing address;

Rule B.4-10: Campus\_Secondary – the secondary contact address for the individual on campus;

Rule B.4-11: Home\_Secondary – the secondary contact home address for the individual;

Rule B.4-12: Mailing\_Secondary – the secondary postal mailing address for the individual;

Rule B.4-13: Permanent\_Secondary – the secondary permanent address for the individual;

Rule B.4-14: Private\_Secondary – the secondary private contact address for the individual;

Rule B.4-15: Temporary\_Secondary – the secondary temporary contact address for the individual;

Rule B.4-16: Work\_Secondary – the secondary work contact address for the individual.

### B1.5 AddressPart Vocabulary

The vocabulary for type of ‘addressPart’ is listed in Table B1.5.

**Table B1.5 Vocabulary for the type of ‘addressPart’.**

Vocabulary Association	Description of the Vocabulary
Type of address: ‘addressPart’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• PoBox;</li> <li>• NonfieldedStreetAddress1;</li> <li>• NonfieldedStreetAddress2;</li> <li>• NonfieldedStreetAddress3;</li> <li>• NonfieldedStreetAddress4;</li> <li>• StreetNumber;</li> <li>• StreetPrefix;</li> <li>• StreetName;</li> <li>• StreetType;</li> <li>• StreetSuffix ;</li> <li>• ApartmentType;</li> <li>• ApartmentNumber;</li> <li>• ApartmentNumberPrefix;</li> <li>• ApartmentNumberSuffix;</li> <li>• Locality;</li> <li>• City;</li> <li>• StatePr;</li> <li>• Region;</li> <li>• Country;</li> <li>• Postcode;</li> <li>• Timezone;</li> <li>• Geo.</li> </ul>

Rule B.5-01: PoBox – post office box number;

Rule B.5-02: NonfieldedStreetAddress1 – unformatted address line 1;

Rule B.5-03: NonfieldedStreetAddress2 – unformatted address line 2;

Rule B.5-04: NonfieldedStreetAddress3 – unformatted address line 3;

Rule B.5-05: NonfieldedStreetAddress4 – unformatted address line 4;

Rule B.5-06: StreetNumber – the house number on the street;

Rule B.5-07: StreetPrefix – the street prefix;

Rule B.5-08: StreetName – the street name;

Rule B.5-09: StreetType – the street type e.g., Road, Crescent etc;

Rule B.5-10: StreetSuffix – the street suffix;

Rule B.5-11: ApartmentType – the apartment type;

Rule B.5-12: ApartmentNumber – the apartment number;

Rule B.5-13: ApartmentNumberPrefix – apartment number prefix;

Rule B.5-14: ApartmentNumberSuffix – the apartment number suffix;

Rule B.5-15: Locality – locality e.g., area within a city;

Rule B.5-16: City – the city name

Rule B.5-17: StatePr – the state/province name;

Rule B.5-18: Region – region within the world e.g., North America;

Rule B.5-19: Country – the country name;

Rule B.5-20: Postcode – the post or zip code;

Rule B.5-21: Timezone – the time zone;

Rule B.5-22: Geo – latitude and longitude in the format defined by ISO 6709. The degrees, minutes and seconds resolution for the combined string of latitude and longitude is given as ‘±DDMMSS.SS±DDMMSS.SS’..

## B1.6 ContactinfoType Vocabulary

The vocabulary for type of ‘contactInfo’ is listed in Table B1.6.

**Table B1.6 Vocabulary for the ‘contactinfoType’.**

Vocabulary Association	Description of the Vocabulary
Type of contact information: ‘contactinfoType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Telephone;</li> <li>• TelephoneHome;</li> <li>• TelephoneWork;</li> <li>• TelephonePrimary;</li> <li>• TelephoneSecondary;</li> <li>• TelephoneHomePrimary;</li> <li>• TelephoneHomeSecondary;</li> <li>• TelephoneWorkPrimary;</li> <li>• TelephoneWorkSecondary;</li> <li>• Facsimile;</li> <li>• FacsimileHome;</li> <li>• FacsimileWork</li> <li>• Mobile;</li> <li>• MobileHome;</li> <li>• MobileWork;</li> <li>• MobileHomePrimary;</li> <li>• MobileWorkPrimary;</li> <li>• MobileHomeSecondary;</li> <li>• MobileWorkSecondary;</li> <li>• Pager;</li> <li>• EmailPrimary;</li> <li>• EmailHomePrimary;</li> <li>• EmailWorkPrimary;</li> <li>• EmailSecondary;</li> <li>• EmailHomeSecondary;</li> <li>• EmailWorkSecondary;</li> <li>• EmailPersonalPrimary;</li> <li>• EmailPersonalSecondary;</li> </ul>

Vocabulary Association	Description of the Vocabulary
	<ul style="list-style-type: none"> <li>• EmailSchoolPrimary;</li> <li>• EmailSchoolSecondary;</li> <li>• Web-Address;</li> <li>• Instant-Message;</li> <li>• SMS.</li> </ul>

Rule B.6-01: Telephone – telephone number;

Rule B.6-02: TelephoneHome – home telephone number;

Rule B.6-03: TelephoneWork – work telephone number;

Rule B.6-04: TelephonePrimary – primary telephone number;

Rule B.6-05: TelephoneSecondary – secondary telephone number;

Rule B.6-06: TelephoneHomePrimary – primary home telephone number;

Rule B.6-07: TelephoneHomeSecondary – secondary home telephone number;

Rule B.6-08: TelephoneWorkPrimary – primary work telephone number;

Rule B.6-09: TelephoneWorkSecondary – secondary work telephone number;

Rule B.6-10: Facsimile – facsimile number;

Rule B.6-11: FacsimileHome – home facsimile number;

Rule B.6-12: FacsimileWork – work facsimile number;

Rule B.6-13: Mobile – mobile telephone number;

Rule B.6-14: MobileHome – home mobile telephone number;

Rule B.6-15: MobileWork – work mobile telephone number;

Rule B.6-16: MobileHomePrimary – primary home mobile telephone number;

Rule B.6-17: MobileWorkPrimary – primary work mobile telephone number;

Rule B.6-18: MobileHomeSecondary – secondary home mobile telephone number;

Rule B.6-19: MobileWorkSecondary – secondary work mobile telephone number;

Rule B.6-20: Pager – pager number;

Rule B.6-21: EmailPrimary – primary email address;

Rule B.6-22: EmailHomePrimary – primary home email address;

Rule B.6-23: EmailWorkPrimary – primary work email address;

Rule B.6-24: EmailSecondary – secondary email address;

Rule B.6-25: EmailHomeSecondary – secondary home email address;

Rule B.6-26: EmailWorkSecondary – secondary work email address;

Rule B.6-27: EmailPersonalPrimary – personal primary email address;

Rule B.6-28: EmailPersonalSecondary – personal secondary email address;

Rule B.6-29: EmailSchoolPrimary – primary school email address;

Rule B.6-30: EmailSchoolSecondary – secondary school email address;

Rule B.6-31: Web-Address – web address;

Rule B.6-32: Instant-Message – instant messaging address;

Rule B.6-33: SMS – SMS number.

### **B1.7 DemographicsType Vocabulary**

The vocabulary for type of 'demographicsType' is listed in Table B1.7.

**Table B1.7 Vocabulary for the 'demographicsType'.**

Vocabulary Association	Description of the Vocabulary
Type of demographic: 'demographicsType'	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Adult;</li> <li>• College;</li> <li>• ContinuingEducation;</li> <li>• Enrichment;</li> <li>• Graduate;</li> <li>• Mature;</li> <li>• Nursery;</li> <li>• Preschool;</li> <li>• Primary;</li> <li>• Professional;</li> <li>• Secondary;</li> <li>• Technical;</li> <li>• University;</li> <li>• Vocational;</li> <li>• Doctoral;</li> <li>• Tertiary;</li> <li>• Residency;</li> <li>• Post Doctoral.</li> </ul>

Rule B.7-01: Adult – demographics information about an adult education participant at the institution

Rule B.7-02: College – college attendance assigned as the context for the demographics information;

Rule B.7-03: ContinuingEducation – information about a continuing education participant at the institution;

Rule B.7-04: Enrichment – demographics information about an enrichment participant at the institution;

Rule B.7-05: Graduate – demographics information about a graduate of the institution;

Rule B.7-06: Mature – demographics information about a mature student of the institution;

Rule B.7-07: Nursery – nursery attendance assigned as the context for the demographics information;

Rule B.7-08: Preschool – preschool attendance assigned as the context for the demographics information;

Rule B.7-09: Primary – primary school attendance assigned as the context for the demographics information;

Rule B.7-10: Professional – demographics information about a professional training at the institution;

Rule B.7-11: Secondary – secondary school attendance assigned as the context for the demographics information;

Rule B.7-12: Technical – demographics information about technical training at the institution;

Rule B.7-13: University – higher educational institution assigned as the context for the demographics information;

Rule B.7-14: Vocational – demographics information about a vocational student at the institution;



Rule B.7-15: Doctoral – demographics information about a doctorate researcher at the institution;

Rule B.7-16: Tertiary – tertiary school attendance assigned as the context for the demographics information;

Rule B.7-17: Residency – demographics information about a medical residency participant at the institution;

Rule B.7-18: Post Doctoral – demographics information about a post-doctoral researcher at the institution.

### **B1.8 DemographicInfo Vocabulary**

The vocabulary for type of ‘demographicInfo’ is listed in Table B1.8.

**Table B1.8 Vocabulary for the ‘demographicInfo’.**

<b>Vocabulary Association</b>	<b>Description of the Vocabulary</b>
Type of vocabulary-based demographic field: ‘demographicInfo’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• PlaceofBirth</li> <li>• MaritalStatus</li> <li>• Ethnicity</li> <li>• Nationality</li> </ul>

Rule B.8-01: PlaceofBirth – the place of birth (the level of detail will vary from just a City to a full address);

Rule B.8-02: MaritalStatus – the marital status (enumerated as Single, Married, Divorced, Widowed and Unknown);

Rule B.8-03: Ethnicity – the ethnicity (there is no defined vocabulary as there are many ways to describe ethnicity);

Rule B.8-04: Nationality – the nationality (there is no defined vocabulary).

### **B1.9 EventDate Vocabulary**

The vocabulary for type of ‘eventDate’ is listed in Table B1.9.

**Table B1.9 Vocabulary for the ‘eventDate’.**

<b>Vocabulary Association</b>	<b>Description of the Vocabulary</b>
Type of event date: ‘eventDate’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Award;</li> <li>• Birth;</li> <li>• Create;</li> <li>• Death;</li> <li>• Delete;</li> <li>• Effective;</li> <li>• Enroll;</li> <li>• Expiry;</li> <li>• Finish;</li> <li>• Join;</li> <li>• Publish;</li> <li>• Renewal;</li> <li>• Start;</li> <li>• Update;</li> <li>• Graduate;</li> <li>• Expel;</li> </ul>

Vocabulary Association	Description of the Vocabulary
	<ul style="list-style-type: none"> <li>• Withdraw;</li> <li>• Military Service.</li> </ul>

Rule B.09-01: Award – date of award;

Rule B.09-02: Birth – date of birth;

Rule B.09-03: Create – date of creation of data;

Rule B.09-04: Death – date of death;

Rule B.09-05: Delete – data deletion date;

Rule B.09-06: Effective – date at which the data becomes effective;

Rule B.09-07: Enroll – date of enrollment;

Rule B.09-08: Expiry – date of expiry of the entry, activity, etc;

Rule B.09-09: Finish – date of completion;

Rule B.09-10: Join – date of joining the associated activity, course, etc;

Rule B.09-11: Publish – date of publication of the data;

Rule B.09-12: Renewal – date of renewal for the data;

Rule B.09-13: Start – start date for the entry, activity, etc;

Rule B.09-14: Update – date of update of the data entry;

Rule B.09-15: Graduate – date of graduation;

Rule B.09-16: Expel – date of expulsion from the institution;

Rule B.09-17: Withdraw – date of withdrawal from course, activity, etc;

Rule B.09-18: Military Service – date of military service.

### **B1.10 RepresentationType Vocabulary**

The vocabulary for type of ‘representationType’ is listed in Table B1.10.

**Table B1.10 Vocabulary for the ‘representationType’.**

Vocabulary Association	Description of the Vocabulary
Type of representation: ‘representationType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Photo;</li> <li>• Voice;</li> <li>• Biometric;</li> <li>• AnalogueSignature;</li> <li>• DigitalSignature.</li> </ul>

Rule B.10-01: Photo – a photograph of the corresponding individual;

Rule B.10-02: Voice – a voice-print of the corresponding individual (this is an audio recording);

Rule B.10-03: Biometric – a biometric representation e.g., retina scan, etc;

Rule B.10-04: AnalogueSignature – a digital representation, scanned, of the individual’s signature;

Rule B.10-05: DigitalSignature – a digital signature, using for example their public encryption key, for the individual.

### **B1.11 AgentType Vocabulary**

The vocabulary for type of ‘agentType’ is listed in Table B1.11.

**Table B1.11 Vocabulary for the ‘agentType’.**

<b>Vocabulary Association</b>	<b>Description of the Vocabulary</b>
Type of agent: ‘agentType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Parent;</li> <li>• Guardian;</li> <li>• Proxy;</li> <li>• Aide;</li> <li>• Advisor;</li> <li>• Tutor;</li> <li>• Mentor;</li> <li>• Sponsor;</li> <li>• Relative.</li> </ul>

Rule B.11-01: Parent – parent of a student;

Rule B.11-02: Guardian – guardian, but not a parent, of a student;

Rule B.11-03: Proxy – someone who can act as a proxy for the student, faculty, etc ;

Rule B.11-04: Aide – an aide, in one capacity or other, for the student, faculty etc;

Rule B.11-05: Advisor – an advisor, in one capacity or another, to the student, faculty, etc;

Rule B.11-06: Tutor – an allocated tutor for a student;

Rule B.11-07: Mentor – an allocated mentor for a student, faculty, etc;

Rule B.11-08: Sponsor – a sponsor, not necessarily in the financial sense, for the student, faculty, etc;

Rule B.11-09: Relative – a relative of the student, faculty, etc.

### **B1.12 EnterpriserolesType Vocabulary**

The vocabulary for type of ‘enterpriserolesType’ is listed in Table B1.12.

**Table B1.12 Vocabulary for the ‘enterpriserolesType’.**

<b>Vocabulary Association</b>	<b>Description of the Vocabulary</b>
Type of role within the enterprise: ‘enterpriserolesType’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• StudentInformationSystem;</li> <li>• HumanResourcesSystem;</li> <li>• Unknown;</li> <li>• Other.</li> </ul>

Rule B.12-01: StudentInformationSystem – for access to the student information system (SIS);

Rule B.12-02: HumanResourcesSystem – for access to the human resources (HR) system;

Rule B.12-03: Unknown – to be used when the role is unknown but general access is required to the system;

Rule B.12-04: Other – to be used when the enterprise system role is known, but no defined in the vocabulary.

### B1.13 SystemRole Vocabulary

The vocabulary for type of ‘systemRole’ is listed in Table B1.13.

**Table B1.13 Vocabulary for the ‘systemRole’.**

Vocabulary Association	Description of the Vocabulary
Type of role within the system: ‘systemRole’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• SysAdmin;</li> <li>• SysSupport;</li> <li>• Creator;</li> <li>• AccountAdmin;</li> <li>• User;</li> <li>• Administrator;</li> <li>• None.</li> </ul>

Rule B.13-01: SysAdmin – to be allocated to a System Administrator;

Rule B.13-02: SysSupport – to be allocated to someone responsible for System Support;

Rule B.13-03: Creator –to be allocated to someone responsible for the creation of user accounts, key data, etc;

Rule B.13-04: AccountAdmin – to be allocated to a user responsible for administration of user accounts;

Rule B.13-05: User – a generic user of the system;

Rule B.13-06: Administrator – to be allocated to someone with a general Administrative role within the institution;

Rule B.13-07: None – to be used when general access is required but there is no specific role.

### B1.14 Institutionrolevalue Vocabulary

The vocabulary for type of ‘institutionrolevalue’ is listed in Table B1.14.

**Table B1.14 Vocabulary for the ‘institutionrolevalue’.**

Vocabulary Association	Description of the Vocabulary
Type of role within the institution: ‘institutionrolevalue’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• Student;</li> <li>• Faculty;</li> <li>• Member;</li> <li>• Learner;</li> <li>• Instructor;</li> <li>• Mentor;</li> <li>• Staff;</li> <li>• Alumni;</li> <li>• ProspectiveStudent;</li> </ul>

Vocabulary Association	Description of the Vocabulary
	<ul style="list-style-type: none"> <li>• Guest;</li> <li>• Other;</li> <li>• Administrator;</li> <li>• Observer;</li> <li>• None.</li> </ul>

Rule B.14-01: Student – assigned to an individual who is a student;

Rule B.14-02: Faculty – assigned to an individual who is a member of the faculty;

Rule B.14-03: Member – assigned to an individual who is a member of the institution (not identifiable by one of the other entries in the vocabulary);

Rule B.14-04: Learner – assigned to an individual who undertaking general learning thru the institution;

Rule B.14-05: Instructor – assigned to an individual who is an instructor;

Rule B.14-06: Mentor – assigned to an individual who is a mentor;

Rule B.14-07: Staff – assigned to an individual who is a member of staff but not faculty or an administrator;

Rule B.14-08: Alumni – assigned to an individual who is an alumni of the institution;

Rule B.14-09: ProspectiveStudent – assigned to an individual who is a prospective student;

Rule B.14-10: Guest – assigned to an individual who is to be given guest access;

Rule B.14-11: Other – assigned to an individual who has a defined role that is not named in the vocabulary;

Rule B.14-12: Administrator – assigned to an individual who is an Administrator in the institution;

Rule B.14-13: Observer – assigned to an individual who is acting as an observer;

Rule B.14-14: None – assigned to an individual who requires general access but has no defined role within the institution.

### B1.15 Enrollment Vocabulary

The vocabulary for type of ‘enrollment’ is listed in Table B1.15.

**Table B1.15 Vocabulary for the ‘systemRole’.**

Vocabulary Association	Description of the Vocabulary
Type of enrollment within the system: ‘enrollment’	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> <li>• AcademicDegree;</li> <li>• AcademicMajor;</li> <li>• AcademicMinor;</li> <li>• AcademicTitle.</li> </ul>

Rule B.15-01: AcademicDegree – the degree on which the learner is enrolled;

Rule B.15-02: AcademicMajor – the academic major subject being studied;

Rule B.15-03: AcademicMinor – the academic minor being studied;

Rule B.15-04: AcademicTitle – the academic title within the institution.

### B1.16 FieldType Vocabulary

The vocabulary for type of field in 'FieldType' is listed in Table B1.16.

**Table B1.16 Vocabulary for the 'FieldType'.**

Vocabulary Association	Description of the Vocabulary
Type of 'ExtensionField': 'FieldType'	<p>The core vocabulary is:</p> <ul style="list-style-type: none"><li>• Boolean;</li><li>• DateTime;</li><li>• Decimal;</li><li>• Integer;</li><li>• String.</li></ul>

Rule B.16-01: Boolean – the data-type is equivalent to the definition of a 'Boolean' in XML;

Rule B.16-02: DateTime – the data-type is equivalent to the definition of a 'DateTime' in XML;

Rule B.16-03: Integer – the data-type is equivalent to the definition of an 'Integer' in XML;

Rule B.16-04: Decimal – the data-type is equivalent to the definition of a 'Decimal' in XML;

Rule B.16-05: String – the data-type is equivalent to the definition of a 'String' in XML.

### B1.17 Language Vocabulary

The language code/country code combination used to identify the language for a piece of text is an enumerated external IMS GLC vocabulary that captures the full set of entries from RFC4646.

## B2 Using Vocabularies for the Metadata Class

The Metadata class consists of attributes:

- `metadataNameVocabulary` – identifies the vocabulary that contains the reference set of `fieldName` values for the meta-data<sup>3</sup>;
- `metadataTypeVocabulary` – identifies the vocabulary that contains the reference set of `fieldType` values for the meta-data. The value for this attribute is the same as the ‘`vocabIdentifier`’ of the VDEX instance for this vocabulary;
- `metadataField` – contains the set of triples (`fieldName`/`fieldType`/`fieldValue`) for each meta-data entry.

The value in the ‘`fieldName`’ must be from the vocabulary (identified using the `metadataNameVocabulary` attribute). The value in the ‘`fieldType`’ must be from the external vocabulary containing the permitted set of external field types (as listed in the `metadataTypeVocabulary` attribute). The value in the ‘`fieldValue`’ is the metadata value itself. Nested values are possible using a dot notation in the ‘`fieldName`’ e.g., for LOM this could be ‘`general.keyword.string`’, etc.

## B3 Using Vocabularies for the IMSExtension Class

The IMSExtension class consists of attributes:

- `extensionNameVocabulary`<sup>4</sup> – identifies the vocabulary that contains the reference set of `fieldName` values for the extension;
- `extensionTypeVocabulary` – identifies the vocabulary that contains the reference set of `fieldType` values for the extension. The value for this attribute is the same as the ‘`vocabIdentifier`’ of the VDEX instance for this vocabulary;
- `extensionField` – contains the set of triples (`fieldName`/`fieldType`/`fieldValue`) for each extension.

The value in the ‘`fieldName`’ must be from the vocabulary (identified using the `extensionNameVocabulary` attribute). The value in the ‘`fieldType`’ must be from the external vocabulary containing the permitted set of external field types (as listed in the `extensionTypeVocabulary` attribute). The value in the ‘`fieldValue`’ is the extension value itself. Nested values are possible using a dot notation in the ‘`fieldName`’ cf. for meta-data.

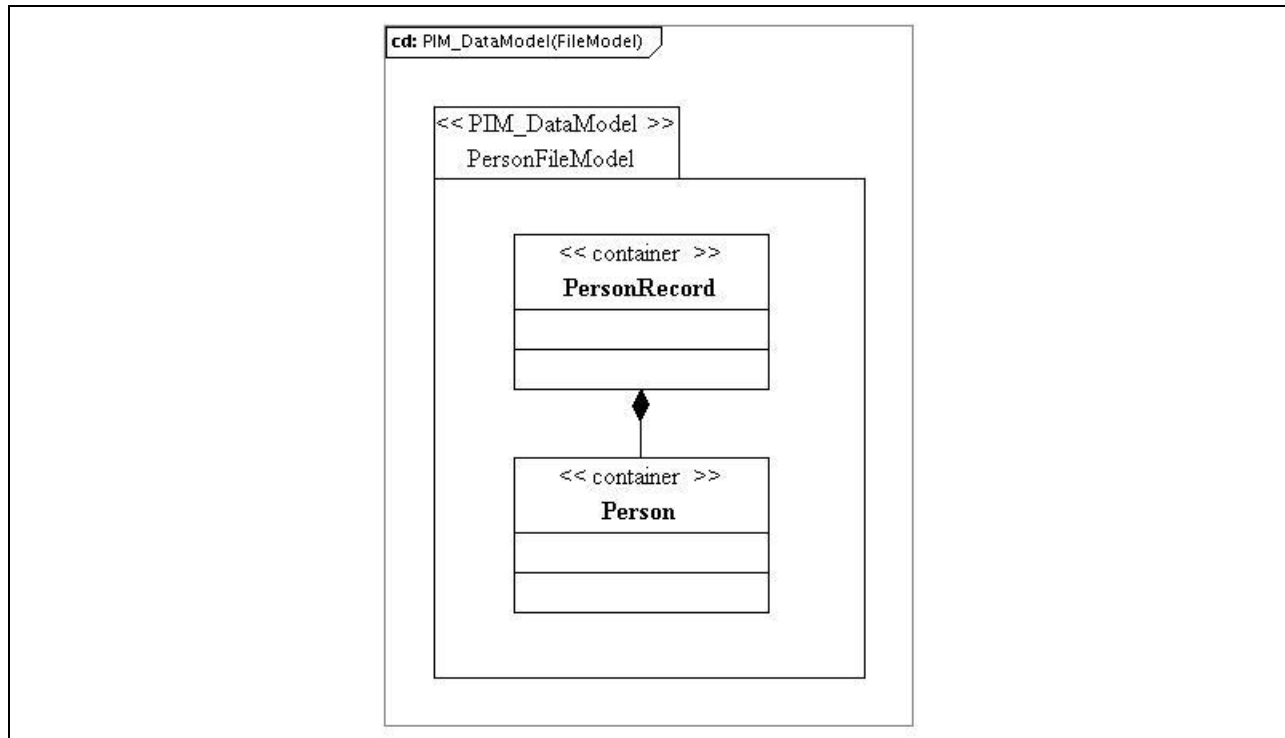
---

<sup>3</sup> The corresponding vocabulary must be defined. It is recommended that the vocabulary registered with IMS GLC made available as a VDEX file. If the vocabulary is defined as a VDEX file then the value for ‘`metadataNameVocabulary`’ should be the ‘`vocabIdentifier`’ of the VDEX instance.

<sup>4</sup> The corresponding vocabulary must be defined. It is recommended that the vocabulary registered with IMS GLC made available as a VDEX file. If the vocabulary is defined as a VDEX file then the value for ‘`extensionNameVocabulary`’ should be the ‘`vocabIdentifier`’ of the VDEX instance.

## Appendix C – File-based Data Exchange

The IMS GLC Bulk Data Exchange Management Service [BDEMS, 11] is used to exchange bulk Membership and related information. The Group information is exchanged by placing multiple *PersonRecord* structures within a bulk data container. Figure C1.1 shows the key class structure and the associated definitions are provided in Section 5 of this document.



**Figure C.1 PersonRecord class diagram for file-based data exchange.**

Note that separate binding instance validation files will be used for containing the description of *PersonRecords* within a file. This ensures that only the required data structures are contained in the corresponding binding validation files.



## About This Document

<b>Title:</b>	IMS GLC Person Management Service Information Model
<b>Editor:</b>	Colin Smythe (IMS GLC)
<b>Co-chairs:</b>	Linda Feng (Oracle) and Bill Lee (Desire2Learn)
<b>Version:</b>	2.0
<b>Version Date:</b>	30 June 2011
<b>Release:</b>	Final 1.0
<b>Status:</b>	<b>Final Release</b>
<b>Summary:</b>	This document contains the IMS GLC Person Management Service v2.0 Information Model. This service is used to exchange information about individuals including name, address, etc. The business transactions include the simple create, read, update and delete of the Person data model for a single instance. The corresponding data model from the IMS GLC Learner Information Package (LIP) v1.0 specification has been combined with the original data model in version 1 of the IMS GLC Person Management Service specification. This document contains the definition of the abstract application-programming interface for the Person Management Service.
<b>Revision Information:</b>	This version supersedes the IMS GLC Person Management Service v1.0 specification.
<b>Purpose:</b>	This document is made available for adoption by the public community at large.
<b>Document Location:</b>	IMS GLC Document Library.

## List of Contributors

The following individuals contributed to the development of this document:

Kerry Blinco	DEEWR (Australia)	Zack Leavitt	Pearson (USA)
Kirk Bunte	SungardHE (USA)	Bill Lee	Desire2Learn (Canada)
Angus Chan	Desire2Learn (Canada)	Richard Moon	SungardHE (USA)
Adam Cooper	JISC/JISC-CETIS (UK)	Mike Parkhill	Desire2Learn (Canada)
Michael Feldstein	Oracle (USA)	Colin Smythe	IMS GLC (UK)
Linda Feng	Oracle (USA)	Reinhold Staudinger	Blackboard (USA)
Chris Hatton	Pearson (USA)	Nick Terrible	University of Wisconsin (USA)
Jon Fontaine	Blackboard (USA)	Jason Zhong	SungardHE (USA)
Karen Kuffner	University of Michigan (USA)		

## Revision History

Version No.	Release Date	Comments
Final Release v1.0	30 June 2011	The first formal release of the Final Release version of this document.

# Index

## A

Abstract Framework ..... 10, 12  
 API..... 12, 14, 16, 17  
 Attributes  
   Address  
     country ..... 94, 102  
     locality ..... 60, 94  
     region..... 94  
     street ..... 93  
   Common  
     dataSource.....7, 49, 51  
     email..... 1, 50, 60, 95  
     extensionField.....8, 84, 85, 103  
     metadataField ..... 103  
     sourcedId . 6, 7, 15, 19, 20, 22, 24, 25, 26, 31, 33, 36, 37, 40, 47, 48  
     userId..... 6, 8, 40, 69, 72  
   Course  
     association.....46  
     section .....42  
   Demographics gender... 7, 61, 63  
   Description  
     FullDescription ..... 4, 8, 81, 82  
   ExtensionField  
     fieldName..... 9, 77, 79, 84, 85, 103  
     fieldType..... 9, 84, 86, 102, 103  
     fieldValue .....9, 86, 103  
   LangString  
     language..8, 35, 42, 54, 56, 58, 60, 64, 67, 68, 71, 72, 73, 74, 77, 78, 79, 80, 82, 83, 85, 86, 102  
     text ..... 80, 81, 82, 83, 102  
   Membership  
     membership..... 24, 36  
   Name  
     nameType. 7, 9, 55, 56, 90  
   Person  
     dataSource.....7, 49, 51  
     email..... 1, 50, 60, 95  
     systemRole .....8, 9, 69, 70, 100, 101  
   personRecord .... 7, 20, 22, 25, 31, 33, 46  
   RecordMetaData  
     comments..... 1, 43

StatusInfo  
   description .10, 11, 14, 64, 65, 66, 68, 80, 81, 82, 104  
 UserId  
   authentication.....10, 74  
   pwEncryption ..... 8, 73  
   userIdType..... 8, 73  
   userIdValue..... 8, 72  
 Values  
   list....11, 54, 56, 58, 60, 62, 64, 65, 67, 70, 71, 75, 77, 86, 88, 91

## B

Binding technologies  
   LDAP ..... 74  
   SOAP ..... 27, 28, 29, 30, 35  
   WSDL ..... 10, 12, 13, 42  
 Bulk Data Exchange  
   Management Service... 10, 12, 104

## C

Classes  
   Common  
     DataSource..... 3, 51  
     ExtensionField.. 4, 8, 9, 85, 86, 102  
     IMSExtension .... 4, 5, 8, 52, 84, 85, 87, 103  
     Metadata ..... 5, 85, 103  
     StatusInfo..... 3, 20, 22, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 41, 88  
     Text..... 4, 8, 42, 54, 60, 67, 68, 72, 73, 74, 77, 78, 79, 80, 82, 83  
     UserId.....3, 4, 8, 40, 72, 73, 74  
   Course  
     Association.90, 91, 92, 93, 94, 96, 97, 98, 99, 100, 101, 102  
     Section.....20, 39, 43, 104  
     FullDescription.....4, 8, 81, 82  
   Name  
     PartName..... 3, 5, 56, 91  
   Person 1, 2, 3, 6, 7, 10, 11, 12, 13, 14, 15, 18, 19, 20, 22, 24, 25, 26, 27, 28, 29, 30,

31, 33, 35, 36, 37, 38, 39, 40, 41, 42, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 59, 61, 66, 69, 71, 74, 75, 87, 89, 105  
 Address.. 3, 6, 7, 50, 57, 58  
 Demographics .. 3, 6, 7, 50, 61, 62, 63, 64  
 FormName ..3, 6, 7, 39, 40, 49, 53, 54  
 Name..... 3, 6, 7, 20, 22, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 49, 55, 56, 87  
 PersonCore ..... 3, 6, 26, 39, 40, 53, 72  
 PersonDatabase3, 6, 7, 45, 46  
 PersonRecord 3, 7, 20, 22, 25, 31, 33, 39, 45, 46, 47, 49, 104  
 Photo .....98  
 Common Services .....10  
 Conformance .....14  
 Course .....10  
 Course Management Service..10  
 Course Structures  
   Association 90, 91, 92, 93, 94, 96, 97, 98, 99, 100, 101, 102  
   Section.....20, 39, 43, 104

## G

Group Management Service ...10

## I

InstitutionRole....4, 8, 71, 74, 75  
 Interface Class  
   PersonManager 2, 6, 9, 18, 19, 88

## L

LDAP .....74  
 Learning Information Services  
   ..... 10, 12  
 Lightweight Directory Access  
   Protocol .....74  
 LIS  
   Bulk Data Exchange  
     Management Service .....10, 12, 104

Course Management Service ..... 10	replacePerson.....6, 19, 20, 22, 31, 33, 37, 38	fullsuccess .20, 22, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 88
Group Management Service ..... 10	updatePerson.....6, 19, 20, 22, 31, 38	idallocfail.....22, 88
Membership Management Service ..... 10	Outcomes Management Service ..... 10	idallocinusefail .....20, 36, 88
Outcomes Management Service ..... 10	<b>P</b>	incompletedata 20, 22, 25, 26, 31, 33, 88
Person Management Service1, 2, 10, 11, 12, 13, 14, 15, 42, 89, 105	Person Management Service1, 2, 10, 11, 12, 13, 14, 15, 42, 89, 105	invaliddata .20, 22, 25, 26, 27, 28, 31, 33, 35, 88
<b>M</b>	Profiling ..... 4, 11, 87	linkfailure .....88, 89
Membership Management Service ..... 10	<b>S</b>	overflowfail .....20, 22, 88
Metadata ..... 5, 85, 103	Services	partialdatastorage ..21, 22, 25, 31, 33, 88
<b>O</b>	Bulk Data Exchange Management..... 10, 12, 104	savepointerror.....28, 30, 88
Operations	Course Management Service ..... 10	savepointsyncerror .28, 30, 88
Person	Group Management ..... 10	targetisbusy .....88, 89
changePersonIdentifier .6, 19, 36, 37	Membership Management.. 10	targetreadfailure .....25, 26, 88
createByProxyPerson.....6, 19, 22, 37	Outcomes Management Service ..... 10	toomuchdata ....27, 28, 29, 30, 35, 88
createPerson 6, 19, 20, 33, 37	Person Management.. 1, 2, 10, 11, 12, 13, 14, 15, 42, 89, 105	unauthorizedrequest .....88, 89
deletePerson .6, 19, 24, 37	SOAP ..... 27, 28, 29, 30, 35	unknownextension.21, 22, 25, 32, 34, 88
discoverPersonIds.....6, 19, 35, 38	Specifications	unknownmdvocabulary20, 22, 25, 29, 30, 31, 33, 88
readPerson ...6, 19, 25, 37, 89	Other	unknownobject 24, 25, 26, 31, 36, 88
readPersonCore ..... 19, 89	LDAP ..... 74	unknownquery .....35, 88
readPersons .6, 19, 29, 38, 89	Status Codes 2, 4, 11, 16, 20, 22, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 88, 89	unknownvocabulary ....20, 22, 31, 33, 88
	createsuccess ..... 33, 88	unsupportedLIS .....88
	deletefailure ..... 24, 88	<b>V</b>
		Vocabularies.....4, 5, 11, 90, 103
		<b>W</b>
		WDSL ..... 10, 12, 13, 42

IMS Global Learning Consortium, Inc. ("IMS GLC") is publishing the information contained in this document ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.

IMS GLC makes no warranty or representation regarding the accuracy or completeness of the Specification.

This material is provided on an "As Is" and "As Available" basis.

The Specification is at all times subject to change and revision without notice.

It is your sole responsibility to evaluate the usefulness, accuracy, and completeness of the Specification as it relates to you.

IMS GLC would appreciate receiving your comments and suggestions.

Please contact IMS GLC through our website at <http://www.imsglobal.org>.

Please refer to Document Name: IMS PMS Information Model v2.0 Final Release v1.0

Date: 30 June 2011