



IMS Enterprise Services Core Use Case Descriptions

Version 1.0 Final Specification

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.msglobal.org/ipr/imsipr_policyFinal.pdf.

Copyright © 2004 IMS Global Learning Consortium. All Rights Reserved.

If you wish to copy or distribute this document, you must complete a valid Registered User license registration with IMS and receive an email from IMS granting the license to distribute the specification. To register, follow the instructions on the IMS website: <http://www.msglobal.org/specificationdownload.cfm>.

This document may be copied and furnished to others by Registered Users who have registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS project group.

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

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.

Copyright © 2004 by IMS Global Learning Consortium, Inc.
All Rights Reserved.

The IMS Logo is a registered trademark of IMS Global Learning Consortium, Inc.

Document Name: IMS Enterprise Services Core Use Case Descriptions

Date: 11 June 2004

Table of Contents

1.	INTRODUCTION	3
1.1	ENTERPRISE SERVICES OVERVIEW	3
1.2	SCOPE AND CONTEXT	3
1.3	STRUCTURE OF THIS DOCUMENT	4
1.4	NOMENCLATURE	4
1.5	REFERENCES	5
2.	PERSON MANAGEMENT SERVICE USE CASES	6
2.1	CREATING A PERSON	6
2.2	READING A PERSON	7
2.3	UPDATING A PERSON	8
2.4	DELETING A PERSON	11
2.5	CHANGING A PERSON OBJECTS SOURCE IDENTIFIER	12
2.6	QUERYING A PERSON	14
2.6.1	<i>Querying Created Persons</i>	<i>14</i>
2.6.2	<i>Querying Updated Persons</i>	<i>15</i>
2.6.3	<i>Querying Deleted Persons</i>	<i>17</i>
3.	GROUP MANAGEMENT SERVICE USE CASES	19
3.1	CREATING A GROUP	19
3.2	READING A GROUP	20
3.3	UPDATING A GROUP	22
3.4	DELETING A GROUP	24
3.5	DELETING A GROUP RELATIONSHIP	26
4.	MEMBERSHIP MANAGEMENT SERVICE USE CASES	29
4.1	CREATING A MEMBERSHIP	29
4.2	READING A MEMBERSHIP	31
4.3	UPDATING A MEMBERSHIP	32
4.4	DELETING A MEMBERSHIP	35
4.5	READING A PERSON'S MEMBERSHIPS	36
4.6	READING A GROUP'S MEMBERSHIP	38
	ABOUT THIS DOCUMENT	40
	LIST OF CONTRIBUTORS	40
	REVISION HISTORY	41
	INDEX	42

1. Introduction

1.1 Enterprise Services Overview

The Enterprise Services specification is the definition of how systems manage the exchange of information that describes people, group and memberships within the context of learning. The Enterprise Services specification is constructed following the recommendations documented in the IMS Abstract Framework (IAF) [AbsGloss, 03], [AbsASC, 03], [AbsWhite, 03]. This means that this specification is based upon:

- Interoperability – Enterprise Services focuses on the exchange of information between Enterprise systems. The specification makes no assumptions on how the data is managed within the Enterprise systems;
- Service-oriented – Enterprise Services defines the exchange of information in terms of the services being supplied by the collaboration of the systems. This takes the form of Person Management Services, Group Management Services and Membership Management Services;
- Component-based – the set of services will be supplied such that they can be combined to form a range of services. The Person Management Services, Group Management Services, and Membership Management Services can be combined to provide other services and the Enterprise Service will have other services added to it in later releases;
- Layering – the Enterprise Service and its constituent services (Person Management, Group Management, and Membership Management) are part of the Application Services layer;
- Behaviors and Data Models – the Enterprise Services 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 Enterprise Services information model is to be defined using the Unified Modelling Language (UML). This enables reliable mapping of the information model into a range of different bindings. The present binding is based upon the Web Services Description Language (WSDL);
- Adoption – the Enterprise Services are based upon the original Enterprise specification data model. While there are significant changes the underlying data model has been maintained and the core Person, Group, and Membership structures remain.

1.2 Scope and Context

This document is the IMS Enterprise Services Core Use Case Descriptions v1.0 and it is used as the basis for the technical requirements addressed in the following documents:

- a) IMS Enterprise Services Specification v1.0 [EntServices, 04a] – the summarized description of the behaviors and accompanying data model for the Person Management, Group Management, and Membership Management services;
- b) IMS Enterprise Services Conformance Specification v1.0 [EntServices, 04b] – the definition of the conformance criteria that must be followed by systems that wish to claim compliance to the Enterprise Services Information model;
- c) IMS Enterprise Services Best Practice & Implementation Guide v1.0 [EntServices, 04c] – this presents information that helps implementers adopt the specification;
- d) IMS Person Management Services Information Model v1.0 [PersonServices, 04a] – the normative Information Model for the Person Management Services;
- e) IMS Person Management Services WSDL Binding v1.0 [PersonServices, 04b] – there are many possible bindings of the Person Management Services Information Model described but at the current time the only specified binding is based upon WSDL;
- f) IMS Group Management Services Information Model v1.0 [GroupServices, 04a] – the normative Information Model for the Group Management Services;

- g) IMS Group Management Services WSDL Binding v1.0 [GroupServices, 04b] – there are many possible bindings of the Group Management Services Information Model described but at the current time the only specified binding is based upon WSDL;
- h) IMS Membership Management Services Information Model v1.0 [MemberServices, 04a] – the normative Information Model for the Membership Management Services;
- i) IMS Membership Management Services WSDL Binding v1.0 [MemberServices, 04b] – there are many possible bindings of the Membership Management Services Information Model described but at the current time the only specified binding is based upon WSDL;

As such the Enterprise Services specification supersedes the original Enterprise specifications:

- a) IMS Enterprise Information Model Final Specification v1.1 [Enterprise, 02a].
- b) IMS Enterprise XML Binding Final Specification v1.1 [Enterprise, 02b];
- c) IMS Enterprise Services Best Practice and Implementation Guide Final Specification v1.1 [Enterprise, 02c].

Within this Core Use Case Descriptions document is the set of use cases against which the Enterprise Services specification has been derived. Not all of the use cases described herein are supported in v1.0 of the Enterprise Services, e.g., querying will be supported in the next release. More use cases will be added as they are identified and so this document should be considered the portfolio for the broad collection of Enterprise Services.

1.3 Structure of this Document

The structure of this document is:

2. Person Management Services Use Cases	The series of core use cases that provide the requirements for a person management service;
3. Group Management Services Use Cases	The series of core use cases that provide the requirements for a group management service;
4. Membership Management Services Use Cases	The series of core use cases that provide the requirements for a membership management service.

1.4 Nomenclature

API	Application Programming Interface
a-API	Abstract Application Programming Interface
CRUD	Create, Read, Update and Delete
HRMS	Human Resources Management System
IAF	IMS Abstract Framework
LMS	Learning Management System
LibMS	Library Management System
MIS	Management Information System
SIF	Schools Interoperability Framework
SIS	Student Information System
TMS	Training Management System
UML	Unified Modelling Language
URL	Universal Resource Locator
VLE	Virtual Learning Environment

W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Mark-up Language

1.5 References

- [AbsASCs, 03] *IMS Abstract Framework: Applications, Services & Components v1.0*, C.Smythe, IMS Global Learning Consortium, Inc., July 2003.
- [AbsGloss, 03] *IMS Abstract Framework: Glossary v1.0*, C.Smythe, IMS Global Learning Consortium, Inc., July 2003.
- [AbsWhite, 03] *IMS Abstract Framework: White Paper v1.0*, C.Smythe, IMS Global Learning Consortium, Inc., July 2003.
- [Cockburn, 01] *Writing Effective Use-case*, A.Cockburn, Addison-Wesley, 2001, ISBN 0-201-70225-8.
- [Enterprise, 02a] *IMS Enterprise Information Model v1.1*, G.Collier, C.Etesse, W.Veres and C.Smythe, IMS Global Learning Consortium, Inc., July 2002.
- [Enterprise, 02b] *IMS Enterprise XML Binding v1.1*, G.Collier, C.Etesse, W.Veres and C.Smythe, IMS Global Learning Consortium, Inc., July 2002.
- [Enterprise, 02c] *IMS Enterprise Best Practice & Implementation Guide v1.1*, G.Collier, C.Etesse, W.Veres and C.Smythe, IMS Global Learning Consortium, Inc., July 2002.
- [EntServices, 04a] *IMS Enterprise Services Specification v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [EntServices, 04b] *IMS Enterprise Services Conformance Specification v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [EntServices, 04c] *IMS Enterprise Services Best Practices and Implementation Guide v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [GroupServices, 04a] *IMS Group Management Services Information Model v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [GroupServices, 04b] *IMS Group Management Services WSDL Binding v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [MemberServices, 04a] *IMS Member Management Services Information Model v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [MemberServices, 04b] *IMS Member Management Services WSDL Binding v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [PersonServices, 04a] *IMS Person Management Services Information Model v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.
- [PersonServices, 04b] *IMS Person Management Services WSDL Binding v1.0*, C.Smythe and C.Vento, IMS Global Learning Consortium, Inc., June 2004.

2. Person Management Service Use Cases

In the context of the following use cases a Person refers to information about an individual. These use cases are documented using the Cockburn [Cockburn, 01] technique.

2.1 Creating a Person

Use Case Title:	Create Person
Use Case Local ID:	Person01/01
Brief Description:	An external HRM system (Reference Agent) wants to transmit newly created employee/student data to the learning system (Sync Agent) so that the learning system's person data will be in sync with the HRM system.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may play the role of a Reference Agent in some instances and in others a Sync Agent).
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the LMS (Sync Agent) has an exact replica of all of the person entries in the HRMS (Reference Agent) and also need to be able to automate this replication to save labor cost.
	Stakeholder: Learners Interest: Each learner from the HRM system needs to have a record present on the LMS in order for them to be able to access the LMS and ensure that records of their training history are properly kept.
Preconditions:	No data dependencies but Reference Agent and Sync Agent must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to create users in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference Agent. All persons in the Reference Agent system have been successfully created on the Sync Agent system.
Triggers:	Can be triggered by the creation of new information on the Reference Agent that needs to be replicated on the Sync Agent(s).
Basic Flow of Events:	1. One or more persons are created on the Reference Agent system; 2. The Reference Agent requests the Sync Agent to create a copy of the supplied person record; 3. The Sync Agent parses the data, creating persons as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.

Use Case Title:	Create Person
Exception Handling:	<ol style="list-style-type: none"> 1. Business-Rule Type Errors <ul style="list-style-type: none"> • An existing Person has the same SourcedId as the object submitted for creation and is judged by the Sync Agent to be a duplicate; 2. Data Errors <ul style="list-style-type: none"> • The data for the object is incomplete (fields required by the Sync Agent are missing) • The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Agents, such as the content of controlled vocabularies) • The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); 3. Infrastructure Errors <ul style="list-style-type: none"> • Authentication error (the identity of the Agents cannot be established) • Authorization error (the Reference Agent is not permitted by the Sync Agent to create a Person object) • Encryption error (the Agents could not decrypt the object) • Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) • Internal error (database connection for example) • Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>Other use cases closely related to this are:</p> <ul style="list-style-type: none"> • Read Person; • Update Person; • Delete Person.
Notes:	None.

2.2 Reading a Person

Use Case Title:	Read Person
Use Case Local ID:	Person01/02
Brief Description:	An LMS (Sync Agent) wants to receive information about a particular person from the external HRM System (Reference Agent) so that the LMSs person data will be in sync with the HRM system
Level:	Sub-function
Actors:	<p>Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.</p> <p>Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent).</p>
Stakeholders & Interest:	<p>Stakeholder: Administrators</p> <p>Interest: Need to make sure the LMS (Sync Agent) have an exact replica of all of the person entries in the HRMS (Reference Agent) and also need to be able to automate this replication to save labor cost.</p>

Use Case Title:	Read Person
	<p>Stakeholder: Learners</p> <p>Interest: Each learner from the HRM system needs to have a record present on the LMS in order for them to be able to access the LMS and ensure that records of their training history are properly kept.</p>
Preconditions:	No data dependencies but source and sync must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to receive data about the specified person in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has received the data record for the specified person.
Triggers:	Can be triggered by a request on the Reference Agent to read data from a Sync Agent so that the validity of the remote data can be ascertained.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. The Sync Agent sends a request for the information about a particular person. The SourcedId is then used to inform the Reference Agent which person's data is being requested. 2. The Sync Agent sends the person record to the Reference Agent as a response to the request 3. The Reference Agent parses the record retrieving the data about the user.
Alternative Flows:	None.
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Sync Agent cannot find a person record with the supplied SourcedId; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to read a Person object) —Encryption error (the Agents could not decrypt the object) <ul style="list-style-type: none"> —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>Other use cases closely related to this are:</p> <ul style="list-style-type: none"> • Update Person; • Delete Person; • Create Person.
Notes:	None.

2.3 Updating a Person

Use Case Title:	Update Person
Use case Local ID:	Person01/03
Brief Description:	An external HRM system (Reference Agent) wants to transmit any employee/student data that has been updated since a given time to the learning system (Sync Agent) so that the LMS's person data will be synchronized with the HRM system.

Use Case Title:	Update Person
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS; The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may play the role of a Reference Agent in some instances and in others a Sync Agent).
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the LMSs (Sync Agents) have an exact replica of all of the person data in the HRMS (Reference Agent) and also need to be able to automate this replication to save labor cost.
	Stakeholder: Learners Interest: Each learner from the HRM system needs to have an accurate and up-to-date record present on the LMS in order for them to be able to access the LMS and ensure that records of their training history are properly kept.
Preconditions:	No data dependencies but the Reference Agent and Sync Agent must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to update person data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference Agent. All data about each person in the Reference Agent system has been successfully updated on the Sync Agent system.
Triggers:	A user updates a Person in the Reference Agent, triggering a request to dependent systems (the Sync Agents) to also update the group. The request trigger may be automatic or manually triggered by an operator.
Basic Flow of Events:	New data values are added to non-repeating data structures:- 1. One or more persons are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified person records with the accompanying data; 3. The Sync Agent parses the data, updating person data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the new data is added.

Use Case Title:	Update Person
Alternative Flows:	<p>Replacement data values are added to non-repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more persons are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified person records with the accompanying data; 3. The Sync Agent parses the data, updating person data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the supplied data replaces that already stored. <p>New data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more persons are updated on the Reference Agent system 2. The Reference Agent sends the Sync Agent a request to update the identified person records with the accompanying data; 3. The Sync Agent parses the data, updating person data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, values provided by the update request are appended to the list of existing values. <p>Replacement data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more persons are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified person records with the accompanying data; 3. The Sync Agent parses the data, updating person data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, the values provided by the update request are used to replace the current values.
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ol style="list-style-type: none"> 1. Business-Rule Type Errors <ul style="list-style-type: none"> • No Person with the given SourcedId can be found to update. Depending on the system, a create may be performed or an exception may be raised; 2. Data Errors <ul style="list-style-type: none"> • The data for the object is incomplete (fields required by the Sync Agent are missing) • The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Agents, such as the content of controlled vocabularies) • The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); 3. Infrastructure Errors <ul style="list-style-type: none"> • Authentication error (the identity of the Agents cannot be established) • Authorization error (the Reference Agent is not permitted by the Sync Agent to update a Person object) • Encryption error (the Agents could not decrypt the object) • Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) • Internal error (database connection for example) • Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>Other use cases closely related to this are:</p> <ul style="list-style-type: none"> • Create Person; • Read person; • Delete Person.
Notes:	None.

2.4 Deleting a Person

Use Case Title:	Delete Person
Use case Local ID:	Person01/04
Brief Description:	An external HRM system (Reference Agent) wants to transmit a record of any Persons deleted since a given time to the learning system (Sync Agent) so that the learning system's person data will be in sync with the HRM system. The stakeholders have an interest in having an exact replica of data from the HRM system, not just a superset, and thus the need for a delete operation.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS; The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may play the role of a Reference Agent in some instances and in others a Sync Agent).
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the LMSs (Sync Agents) have an exact replica of all of the person entries in the HRMS (Reference Agent) and also need to be able to automate this replication to save labor cost.
	Stakeholder: Learners Interest: Learners that don't have access to a record of themselves on the source system or who also do not have access to a record of themselves in the LMS.
Preconditions:	No data dependencies but Reference Agent and Sync Agent must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to delete person data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference Agent. All persons deleted in the Reference Agent system have been deleted on the Sync Agent system.
Triggers:	Can be triggered by a request from the Reference Agent to delete all persons described by the supplied identifiers.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more persons are deleted on the Reference Agent system; 2. The Reference Agent sends the delete person request to the Sync Agent as a response to the request; 3. The Sync Agent parses the data, deleting persons as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.

Use Case Title:	Delete Person
Exception Handling:	<ol style="list-style-type: none"> 1. Business-Rule Type Errors <ul style="list-style-type: none"> • No Person with the same SourcedId as the one provided can be found and thus, the Sync agent is unable to delete anything; 2. Infrastructure Errors <ul style="list-style-type: none"> • Authentication error (the identity of the Agents cannot be established) • Authorization error (the Reference Agent is not permitted by the Sync Agent to delete a Person object) • Encryption error (the Agents could not decrypt the object) • Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) • Internal error (database connection for example) • Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>Other use cases closely related to this are:</p> <ul style="list-style-type: none"> • Read person; • Update Person; • Create Person.
Notes:	None.

2.5 Changing a Person Objects Source Identifier

Use Case Title:	UpdateSourcedID
Use case Local ID:	Person01/05
Brief Description:	An external HR system (Reference Agent) wants to transmit any employee/student data that has been updated since a given time to the learning system (Sync Agent) so that the learning system's person data will be in sync with the HR system
Level:	Sub-function
Actors:	<p>Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS, Management;</p> <p>The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.</p>
	<p>Secondary:</p> <p>Secondary: Sync Agent – generalized from LMS, VLE, LibLMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems.</p> <p>(A system may sometimes play the role of a Source Agent in some instances and in others a Sync Agent depending on the situation.)</p>
Stakeholders & Interest:	<p>Stakeholder: Administrators</p> <p>Interest: Need to make sure the learning (Sync) systems have an exact replica of all of the person data in the HR (Reference) system and also needs to be able to automate this replication to save labor cost</p>
	<p>Stakeholder: Learners</p> <p>Interest: Each learner from the HR system needs to have an accurate and up-to-date record present on the learning system in order for them to be able to access the learning system and ensure that records of their training history are properly kept</p>

Use Case Title:	UpdateSourcedID
Preconditions:	No data dependencies but Reference Agent and Sync Agent must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to update data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception
Success Guarantees:	The Sync Agent has the same data as the Reference Agent. All data about each person in the Reference Agent system has been successfully updated on the Sync Agent system.
Triggers:	Can be triggered by either a manual request on the sync agent to retrieve all IDs that have been updated since a specific date, or by an automated process on the sync agent that periodically retrieves all IDs updated since a specific date.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more IDs are updated on the Reference Agent system. 2. The Sync Agent requests the Reference Agent send over any IDs updated since a given time 3. The Reference Agent sends the original SourcedId and new SourcedId data to the Sync Agent as a response to the request 4. The Sync Agent parses the xml, updating SourcedIds as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ol style="list-style-type: none"> 1. Business-Rule Type Errors <ul style="list-style-type: none"> • No object with the given SourcedId can be found to update. 2. Data Errors <ul style="list-style-type: none"> • The data for the object is incomplete (fields required by the Sync Agent are missing) • Depending on the requirements of Agents, the data required for completion may be <i>minimal</i> (required identifiers only, with subsequent Update Person actions filling in the values as needed), or it may be <i>comprehensive</i> (all values complete on creation) • The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Sync Agent, such as the content of controlled vocabularies) • The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent) 3. Infrastructure Errors <ul style="list-style-type: none"> • Authentication error (the identity of the Sync Agent cannot be established) • Encryption error (the Sync Agent could not decrypt the object) • Signing error (the Sync Agent could not guarantee the data had not been changed between transmission and receipt) • Internal error (database connection for example) • Communication error (the message itself could not be correctly interpreted)
Extensions Points:	None.
Notes:	None.

2.6 Querying a Person

2.6.1 Querying Created Persons

Use Case Title:	Querying a Person using Creation Date
Use case Local ID:	Person02/01
Brief Description:	The learning system (Sync Agent) wishes to synchronize with the HRM system (Reference Agent). It needs to acquire any newly created employee/student data from the Reference Agent.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LMS and portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Source Agent in some instances and in others a Sync Agent depending on the situation).
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the learning (Sync) systems have an exact replica of all of the person entries in the HRMS (Reference) and also need to be able to automate this replication to save labor cost.
	Stakeholder: Learners Interest: Each learner from the HRMS needs to have a record present on the learning system in order for them to be able to access the learning system and ensure that records of their training history are properly kept.
Preconditions:	No data dependencies but reference and sync must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to receive data about the specified person in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference agent. All persons in the Reference Agent system have been successfully created on the Sync Agent system.
Triggers:	Can be triggered by either a manual request on the Sync Agent to retrieve all persons that have been created since a specific date, or by an automated process on the Sync Agent that periodically retrieves all persons created since a specific date.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more persons are created on the Reference Agent system; 2. The Sync Agent requests the source agent to send over any persons created since a given time i.e., it issues a query on all 'person' records after a particular date of creation; 3. The Reference Agent sends the person records to the Sync Agent as a response to the request; 4. The Sync Agent parses the data, processing the person records as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.

Use Case Title:	Querying a Person using Creation Date
Postconditions:	The Sync Agent has received a response from the Reference Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ul style="list-style-type: none"> Business-Rule Type Errors <ul style="list-style-type: none"> —An invalid Query has been requested; Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Sync Agent are missing) —Depending on the requirements of the Agents, the data may be required for completion may be <i>minimal</i> (required identifiers only, with subsequent Update Person actions filling in the values as needed), or it may be <i>comprehensive</i> (all values complete on creation) —The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Sync Agent, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to query a Person object) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	Other use cases closely related to this are: <ul style="list-style-type: none"> Update Person.
Notes:	None.

2.6.2 Querying Updated Persons

Use Case Title:	Query a Person using Update Date
Use case Local ID:	Person02/02
Brief Description:	The learning system (Sync Agent) wishes to synchronize with the HRM system (Reference Agent). It needs to acquire any recently updated employee/student data from the Reference Agent.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation).

Use Case Title:	Query a Person using Update Date
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the learning (Sync) systems have an exact replica of all of the person data in the HRMS (Reference) and also need to be able to automate this replication to save labor cost
	Stakeholder: Learners Interest: Each learner from the HRMS needs to have an accurate and up-to-date record present on the learning system in order for them to be able to access the learning system and ensure that records of their training history are properly kept
Preconditions:	No data dependencies but source and sync must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to receive data about the specified person in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference Agent. All data about each person record in the Reference Agent system has been successfully updated on the Sync Agent system.
Triggers:	Can be triggered by either a manual request by the Sync Agent to retrieve all person records that have been updated since a specific date, or by an automated process on the Sync Agent that periodically retrieves all persons updated since a specific date.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more person records are updated on the Reference Agent system; 2. The Sync Agent requests the Reference Agent send over any person records updated since a given time i.e., it issues a query on all 'person' records updated after a particular time; 3. The source agent sends the person record to the Sync Agent as a response to the request; 4. The Sync Agent parses the data, updating the person records as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Sync Agent has received a response from the Reference Agent indicating either success or the exceptions that were encountered.

Use Case Title:	Query a Person using Update Date
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —An invalid Query has been requested; • Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Sync Agent are missing) —Depending on the requirements of Agents, the data may be required for completion may be <i>minimal</i> (required identifiers only, with subsequent Update Person actions filling in the values as needed), or it may be <i>comprehensive</i> (all values complete on creation). —The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Sync Agent, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to read a Person object) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	Other use cases closely related to this are: <ul style="list-style-type: none"> • Create Person; • Update Person.
Notes:	None.

2.6.3 Querying Deleted Persons

Use Case Title:	Query a Person using Deletion Date
Use case Local ID:	Person02/03
Brief Description:	The learning system (Sync Agent) wishes to synchronize with the HRM system (Reference Agent). It needs to acquire any recently deleted employee/student data from the Reference Agent.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SIS, MIS, HRMS, TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation).
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the learning (Sync) systems have an exact replica of all of the person entries in the HRMS (Reference) and also need to be able to automate this replication to save labor cost.

Use Case Title:	Query a Person using Deletion Date
	<p>Stakeholder: Learners</p> <p>Interest: Learners that don't have access or a record of themselves to the source system must also not have access or a record of themselves in the learning system</p>
Preconditions:	No data dependencies but source and sync must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to receive data about the specified person in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent has the same person data as the Reference Agent. All persons deleted in the Reference Agent system have been deleted created on the Sync Agent system.
Triggers:	Can be triggered by either a manual request on the Sync Agent to retrieve all persons that have been deleted since a specific date, or by an automated process on the Sync Agent that periodically retrieves all persons deleted since a specific date.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more person records are deleted on the Reference Agent system; 2. The Sync Agent requests the Reference Agent sends over any persons deleted since a given time i.e., it issues a query on all 'person' records deleted after a particular time; 3. The Reference Agent sends the person identifiers to the Sync Agent as a response to the request; 4. The Sync Agent parses the identifiers, deleting person records as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Sync Agent has received a response from the Reference Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —An invalid Query has been requested; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to read a Person object) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>Other use cases closely related to this are:</p> <ul style="list-style-type: none"> • Delete Person.
Notes:	None.

3. Group Management Service Use Cases

In these use cases a Group is any collection of Persons or other Groups. A Group is an abstraction of collections such as a set of lectures in a course, a set of individuals in a tutorial, etc.

3.1 Creating a Group

Use Case Title:	Create Group
Use case Local ID:	Group01/01
Brief Description:	A system agent (the Reference Agent) that holds information about the membership of people in groups, requests another system agent (the Sync Agent) to create a new group record so that its information is synchronized.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may play the role of a Reference Agent in some instances and in others a Sync Agent).
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on group existence and inter-group relationships is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
Preconditions:	Group described cannot already exist.
Minimal Guarantees:	The Sync Agent is informed of a new Group object and is requested to create it. Where this request is unsuccessful status information is returned to the Reference Agent to help track down the problem.
Success Guarantees:	The Sync Agent contains the same Group information as the Reference Agent upon completion of the Use Case.
Triggers:	A user creates a new Group for the Reference Agent, triggering a request to a dependent system (the Sync Agent) to also create the group. The request trigger may be automatic – generated by object creation – or it may be triggered by an operator.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system successfully creates a Group on the Reference Agent; 2. The Reference Agent sends a request to the Sync Agent to create a new Group object; 3. The Sync Agent processes the request and attempts to create the object. If a SourcedId was provided for the object, then the object is created with that SourcedId, otherwise, the Sync Agent generates a SourcedId for the new object; 4. The Sync Agent informs the Reference Agent that the object was created. If a SourcedId was <i>not</i> provided for the object, then the Sync Agent returns the identifier that the Sync Agent has given the new Group object to the Reference Agent; 5. The response from the Sync Agent contains a status code that reflects the success or otherwise of the request.

Use Case Title:	Create Group
Alternative Flows:	None.
Postconditions:	The Reference Agent has requested that the Sync Agent creates a Group, and has received a response in return.
Exception Handling:	<p>The Sync Agent must be able to inform the Reference Agent if there are problems creating the new Group record. The Sync Agent processes the request and attempts to create the object, but cannot complete the requested operation, which could be due to:</p> <ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —An existing Group has the same SourcedId(s) as the object submitted for creation and is judged by the Sync Agent to be a duplicate —Other business rule violations have occurred which prevent object creation, for example there may be special rules concerning types of groups that can be created depending on lifecycle or other conditions that would be known to particular Agents; • Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Sync Agent are missing) —The data for the object is incorrect (values of the submitted objects do not match field requirements set for the Agents, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request the creation of Group objects) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Update Group; • Delete Group; • Request Group; • Create Membership (pre-condition).
Notes:	None.

3.2 Reading a Group

Use Case Title:	Read Group
Use case Local ID:	Group01/02
Brief Description:	An external system agent (the Reference Agent) requests another system agent (the Sync Agent) to retrieve and display a group record.
Level:	Sub-function

Use Case Title:	Read Group
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system that has interest in the requested information. This is typically the source system for the information being requested, but not always.
	Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may play the role of a Sync Agent in some instances and in others a Reference Agent depending on the situation).
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about groups held in the MIS/SIS so that resource use is correctly managed.
Preconditions:	Group record referenced by request call exists.
Minimal Guarantees:	The Sync Agent is informed of the request for the group object. Where this request is unsuccessful status information is returned to the Reference Agent to help track down the problem.
Success Guarantees:	The Sync Agent returns the requested group object to the Reference Agent upon completion of the Use Case.
Triggers:	A user or system wishes to validate a previously requested group record on the Sync system. The request trigger may be automatic – generated for auditing purposes – or it may be manually triggered by an operator.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as an automated verification system initiates a validation test suite. 2. The Reference Agent sends a request to the Sync Agent to retrieve the group object. 3. The Reference Agent processes the request and attempts to retrieve the object. 4. The Sync Agent returns the requested object or exception information to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Reference Agent has requested that the Sync Agent retrieve a Group, and has received a response in return.

Use Case Title:	Read Group
Exception Handling:	<p>The Sync Agent must be able to inform the Reference Agent if there are problems retrieving the group record. The error states are:</p> <ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Request object makes reference to a Group that the Sync Agent could not locate with the SourcedIds provided —No existing group has the same SourcedId(s) as the object requested; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request data from group objects) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Create Group; • Update Group; • Delete Group.
Notes:	None.

3.3 Updating a Group

Use Case Title:	Update Group
Use case Local ID:	Group01/03
Brief Description:	A system agent (the Reference Agent) that holds information about groups, requests another system agent (the Sync Agent) to update an existing group record so that its information is synchronized.
Level:	Sub-function
Actors:	<p>Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.</p> <p>Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation).</p>
Stakeholders & Interest:	<p>Stakeholder: Administrative personnel Interest: Need to ensure that information held on groups is consistent across applications, and minimize the need for manual intervention.</p> <p>Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about groups held in the MIS/SRS so that resource use is correctly managed.</p> <p>Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct courses created in the correct relationships to other courses.</p>

Use Case Title:	Update Group
	<p>Stakeholder: Student</p> <p>Interest: Need to have administrative systems stay up to date with their status in the organization regarding courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.</p>
Preconditions:	The Group records referenced are already known to the Reference Agent and have been previously communicated to the Sync Agent.
Minimal Guarantees:	The Sync Agent is informed of updated Group objects and is requested to update it. Where this request is unsuccessful status information is returned to the Reference Agent to help track down the problem.
Success Guarantees:	The Sync Agent contains the same Group information as the Reference Agent upon completion of the Use Case.
Triggers:	A user updates a Group in the Reference Agent, triggering a request to dependent systems (the Sync Agents) to also update the group. The request trigger may be automatic or manually triggered by an operator.
Basic Flow of Events:	<p>New data values are added to non-repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more groups are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified group records with the accompanying data; 3. The Sync Agent parses the data, updating group data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the new data is added.
Alternative Flows:	<p>Replacement data values are added to non-repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more groups are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified person records with the accompanying data; 3. The Sync Agent parses the data, updating group data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the supplied data replaces that already stored. <p>New data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more groups are updated on the Reference Agent system 2. The Reference Agent sends the Sync Agent a request to update the identified group records with the accompanying data; 3. The Sync Agent parses the data, updating group data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, values provided by the update request are appended to the list of existing values. <p>Replacement data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more groups are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified group records with the accompanying data; 3. The Sync Agent parses the data, updating group data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, the values provided by the update request are used to replace the current values.
Postconditions:	The Reference Agent has requested that the Sync Agent update a Group, and has received a response in return.

Use Case Title:	Update Group
Exception Handling:	<p>The Sync Agent must be able to inform the Reference Agent if there are problems update the Group record. The error codes are:</p> <ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —Business rule violations have occurred which prevent object creation or update; for example there may be special rules concerning types of groups that can be created depending on group lifecycle or other conditions that would be known to particular Agents; —No Group with the given SourcedId can be found to update. Depending on the system, a create may be performed or an exception may be raised; • Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Agents are missing) —The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Agents, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Agent); • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agent cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to update data in a Group object) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Create Group; • Delete Group; • Request Group.
Notes:	None.

3.4 Deleting a Group

Use Case Title:	Delete Group
Use case Local ID:	Group01/04
Brief Description:	A system agent (the Reference Agent) that holds information about groups, requests another system agent (the Sync Agent) to delete a group record so that its information is synchronized. On success, the group is removed, as are any memberships in this group.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.

Use Case Title:	Delete Group
	<p>Secondary: Sync Agent – generalized from LMS, VLE, LibMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems.</p> <p>(A system may play the role of a Reference Agent in some instances and in others a Sync Agent).</p>
Stakeholders & Interest:	<p>Stakeholder: Administrative personnel</p> <p>Interest: Need to ensure that information held on groups is consistent across applications, and minimize the need for manual intervention.</p>
	<p>Stakeholder: Library management personnel</p> <p>Interest: Need their system to hold up-to-date information about groups held in the MIS/SRS so that resource use is correctly managed.</p>
	<p>Stakeholder: Tutors</p> <p>Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct courses created in the correct relationships to other courses.</p>
	<p>Stakeholder: Student</p> <p>Interest: Need to have administrative systems stay up to date with their status in the organization regarding courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.</p>
Preconditions:	The Group referenced must already exist
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to delete group data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent contains the same Group information as the Reference Agent upon completion of the Use Case. Any memberships for this group are also removed.
Triggers:	<p>A user deletes a Group for the Reference Agent, triggering a request to dependent systems (the Sync Agent) to also delete the group. The request trigger may be automatic or triggered manually by an operator.</p> <p>The delete Group action will also trigger the deletion of any group or membership objects associated with the deleted group.</p>
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system successfully deletes a Group on the Reference Agent; 2. The Reference Agent sends a request to the Sync Agent to delete the Group object; 3. The Sync Agent processes the request and attempts to delete the object and any associated memberships; 4. The Sync Agent informs the Reference Agent that the object was deleted or reports the appropriate exception code.
Alternative Flows:	None.
Postconditions:	The Reference Agent has requested that the Sync Agent delete a Group, and has received a response in return.

Use Case Title:	Delete Group
Exception Handling:	<p>The Sync Agent must be able to inform the Reference Agent if there are problems deleting the Group record. The error codes are:</p> <ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Group object makes reference to a Group that the Sync Agent could not locate with the SourcedIds provided —Notification that deletion of the Group will cause record consistency errors due to the associated Groups and/or memberships —Other business rule violations have occurred which prevent object deletion, for example there may be special rules concerning types of groups that can must be maintained or other conditions that would be known to particular Agents; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request the deletion of Group objects) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Create Group; • Request Group; • Update Group; • Update Membership.
Notes:	None.

3.5 Deleting a Group Relationship

Use Case Title:	Delete Group Relationship
Use case Local ID:	Group01/05
Brief Description:	<p>An external HRM system (Reference Agent) wants to transmit records of any Group relationships deleted since a given time to the learning system (Sync Agent) so that the learning system's person data will be in sync with the HRM system. The stakeholders have an interest in having an exact replica of data from the HRM system, not just a superset, and thus the need for a delete operation.</p> <p>This use case is intended to support removal of relationships, which is not easily covered via the DeleteGroup and UpdateGroup use cases.</p>
Level:	Sub-function
Actors:	<p>Primary: Source Agent – generalized from SIS, MIS, HRMS and TMS The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.</p>
	<p>Secondary: Sync Agent – generalized from LMS, VLE, LibLMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Source Agent in some instances and in others a Sync Agent depending on the situation.)</p>

Use Case Title:	Delete Group Relationship
Stakeholders & Interest:	Stakeholder: Administrators Interest: Need to make sure the learning (Sync) systems have an exact replica of all of the group entries in the HRM (Reference) system and also needs to be able to automate this replication to save labor cost.
Preconditions:	No data dependencies but Reference Agent and Sync Agent must be able to connect with each other (ability to discover one another not a precondition).
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to delete group attributes in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception
Success Guarantees:	The Sync Agent has the same group data as the Reference Agent. All group relationships deleted in the Reference Agent system have been deleted on the Sync Agent system. Only relationships specified have been removed. All other attributes of group remain unchanged. All mandatory group fields are simply provided for completeness.
Triggers:	Can be triggered by either a manual request on the sync agent to retrieve all group relationships that have been deleted since a specific date, or by an automated process on the sync agent that periodically retrieves all group relationships deleted since a specific date.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. One or more group relationships are deleted on the Reference Agent system; 2. The Sync Agent requests the Reference Agent send over any group attributes deleted since a given time; 3. The Reference Agent sends the group XML data to the Sync Agent as a response to the request; 4. The Sync Agent parses the XML, deleting group attributes as necessary and reporting any exceptions to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Reference Agent has received a response from the Sync Agent indicating either success or the exceptions that were encountered.
Exception Handling:	<ol style="list-style-type: none"> 1. Business-Rule Type Errors <ul style="list-style-type: none"> • No Group relationship with the same value as the one provided can be found and thus, the Sync Agent is unable to delete anything; 2. Data Errors <ul style="list-style-type: none"> • The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Sync Agent, such as the content of controlled vocabularies) • The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); 3. Infrastructure Errors <ul style="list-style-type: none"> • Authentication error (the identity of the Sync Agent cannot be established) • Encryption error (the Sync Agent could not decrypt the object) • Signing error (the Sync Agent could not guarantee the data had not been changed between transmission and receipt) • Internal error (database connection for example) • Communication error (the message itself could not be correctly interpreted).

Use Case Title:	Delete Group Relationship
Extensions Points:	Other use cases closely related to this are: <ul style="list-style-type: none">• Update Group• Delete Group.
Notes:	None.

4. Membership Management Service Use Cases

4.1 Creating a Membership

Use Case Title:	Create Membership
Use case Local ID:	Membership01/01
Brief Description:	A system agent (the Reference Agent) that holds information about the membership of people in groups, requests another system agent (the Sync Agent) to create a new membership record so that its information is synchronized.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation.)
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	Both the Person and Group records referenced by the Membership are already known to the Reference Agent and have been previously communicated to the Sync Agent.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to create membership data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent contains the same Membership information as the Reference Agent upon completion of the Use Case.
Triggers:	A user creates a new Membership for the Reference Agent, triggering a request to a dependent system (the Sync Agent) to also create the membership. The request trigger may be automatic – generated by object creation – or may be manually triggered by an operator.

Use Case Title:	Create Membership
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system successfully creates a Membership on the Reference Agent; 2. The Reference Agent sends a request to the Sync Agent to create a new Membership object; 3. The Sync Agent processes the request and attempts to create the object. If a SourcedId was provided for the object, then the object is created with that SourcedId, otherwise, the Sync Agent generates a SourcedId for the new object; 4. The Sync Agent informs the Reference Agent that the object was created or returns an exception code. If a SourcedId was <i>not</i> provided for the object, then the Sync Agent passes the identifier that the Sync Agent has given the new Membership object to the Reference Agent.
Alternative Flows:	None.
Postconditions:	The Reference Agent has requested that the Sync Agent creates a Membership, and has received a response in return.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Membership object makes reference to a Person or Group that the Sync Agent could not locate with the SourcedIds provided —An existing Membership has the same SourcedId(s) as the object submitted for creation and is judged by the Sync Agent as likely to be a duplicate according to its criteria —Other business rule violations have occurred which prevent object creation; for example there may be special rules concerning types of memberships that can be created depending on group lifecycle or other conditions that would be known to particular Agents; • Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Sync Agent are missing) —The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Agents, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Agent); • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request the creation of Membership objects) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Update Membership; • Delete Membership; • Request Membership.
Notes:	None.

4.2 Reading a Membership

Use Case Title:	Read Membership
Use case Local ID:	Membership01/02
Brief Description:	A system agent (the Sync Agent) that has an interest in information about people and groups, requests another system agent (the Reference Agent) to provide specific membership information.
Level:	Sub-function
Actors:	Primary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation).
	Secondary: Reference Agent – generalized from SRS, MIS, HRMS TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	The Membership is already known to the Reference Agent and identifying information (in the form of a SourcedId) has been previously communicated to the Sync Agent.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to receive data about the specified membership in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent receives the data for the Membership it requested.
Triggers:	A user or process requests information about a specific Membership on the Sync Agent.

Use Case Title:	Read Membership
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system requests information on the Sync Agent about a specific Membership object; 2. The Sync Agent sends a request to the Reference Agent to return the Membership object, using a SourcedId to specify the Membership needed; 3. The Reference Agent processes the request and attempts to access and return the object; 4. The Reference Agent returns the either Membership object to the Sync Agent or the appropriate exception code.
Alternative Flows:	None.
Postconditions:	The Sync Agent has requested that the Reference Agent provides data for a Membership, and has received a response in return.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Reference Agent could not locate a Membership with the SourcedId provided; —Other business rule violations have occurred which prevent object retrieval; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Sync Agent is not permitted by the Reference Agent to request Membership objects) —Encryption error (the Agents could not decrypt the objects) —Signing error (the Agents could not guarantee the request had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Update Membership; • Delete Membership; • Create Membership.
Notes:	None.

4.3 Updating a Membership

Use Case Title:	Update Membership
Use case Local ID:	Membership01/03
Brief Description:	A system agent (the Reference Agent) that holds information about the membership of people in groups, requests another system agent (the Sync Agent) to update an existing membership record so that its information is synchronized.
Level:	Sub-function
Actors:	<p>Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.</p> <p>Secondary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation.)</p>

Use Case Title:	Update Membership
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	The Membership object already exists in both the Reference Agent and Sync Agent.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Reference Agent is able to update data about the specified membership in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent contains the same Membership information as the Reference Agent upon completion of the Use Case.
Triggers:	A user updates an existing Membership on the Reference Agent, triggering a request to dependent systems (the Sync Agent) to also update the membership. The request trigger may be automatic – generated by object update – or manually triggered by an operator.
Basic Flow of Events:	New data values are added to non-repeating data structures:- 1. One or more memberships are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified membership records with the accompanying data; 3. The Sync Agent parses the data, updating membership data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the new data is added.

Use Case Title:	Update Membership
Alternative Flows:	<p>Replacement data values are added to non-repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more memberships are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified membership records with the accompanying data; 3. The Sync Agent parses the data, updating membership data as necessary and reporting any exceptions to the Reference Agent. For the non-repeating fields the supplied data replaces that already stored. <p>New data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more memberships are updated on the Reference Agent system 2. The Reference Agent sends the Sync Agent a request to update the identified membership records with the accompanying data; 3. The Sync Agent parses the data, updating membership data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, values provided by the update request are appended to the list of existing values. <p>Replacement data values are added to repeating data structures:-</p> <ol style="list-style-type: none"> 1. One or more memberships are updated on the Reference Agent system; 2. The Reference Agent sends the Sync Agent a request to update the identified membership records with the accompanying data; 3. The Sync Agent parses the data, updating membership data as necessary and reporting any exceptions to the Reference Agent. For the repeating fields, the values provided by the update request are used to replace the current values.
Postconditions:	The Reference Agent has requested that the Sync Agent updates a Membership, and has received a response in return.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Membership object makes reference to a Person or Group that the Sync Agent could not locate with the SourcedIds provided —The Membership object cannot be located with the given SourcedId —Other business rule violations have occurred which prevent the object being updated; • Data Errors <ul style="list-style-type: none"> —The data for the object is incomplete (fields required by the Sync Agent are missing) —The data for the object is incorrect (values of the submitted objects do not match field requirements set by the Agents, such as the content of controlled vocabularies) —The data for the object is invalid (the data is incorrectly typed or cannot be validated against a schema by the Sync Agent); • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request the updating of Membership objects) —Encryption error (the Agents could not decrypt the update request) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).

Use Case Title:	Update Membership
Extensions Points:	This Use Case is related to the following use cases: <ul style="list-style-type: none"> • Create Membership; • Delete Membership; • Request Membership.
Notes:	None.

4.4 Deleting a Membership

Use Case Title:	Delete Membership
Use case Local ID:	Membership01/04
Brief Description:	A system agent (the Reference Agent) that holds information about the membership of people in groups, requests another system agent (the Sync Agent) to delete an existing membership record so that its information is synchronized.
Level:	Sub-function
Actors:	Primary: Reference Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
	Secondary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Reference Agent in some instances and in others a Sync Agent depending on the situation.)
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	The Membership record is already known to the Reference and Sync Agents.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to delete membership data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent contains the same Membership information as the Reference Agent upon completion of the Use Case.

Use Case Title:	Delete Membership
Triggers:	A user deletes an existing Membership for the Reference Agent, triggering a request to dependent systems (the Sync Agent) to also delete the membership. The request trigger may be automatic – generated by object deletion – or manually triggered by an operator.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system successfully deletes a Membership on the Reference Agent; 2. The Reference Agent sends a request to the Sync Agent to delete the corresponding Membership object, passing its SourcedId; 3. The Sync Agent processes the request and attempts to delete the object. Objects are not necessarily deleted in fact, but in terms of subsequent communication with the Reference Agent the object should be treated as having been deleted (the Reference Agent may actually simply flag the object as inactive in its local implementation rather than removing it); 4. The Sync Agent informs the Reference Agent that the object was deleted.
Alternative Flows:	None.
Postconditions:	The Reference Agent has requested that the Sync Agent deletes a Membership, and has received a response in return.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —A Membership object could not be located with the SourcedId provided —Other business rule violations have occurred which prevent object deletion; for example there may be special rules concerning types of memberships that can be deleted depending on group lifecycle or other conditions that would be known to particular Agents; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Reference Agent is not permitted by the Sync Agent to request the deletion of Membership objects) —Encryption error (the Agents could not decrypt the object) —Signing error (the Agents could not guarantee the data had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	<p>This Use Case is related to the following use cases:</p> <ul style="list-style-type: none"> • Update Membership; • Create Membership; • Request Membership.
Notes:	None.

4.5 Reading a Person's Memberships

Use Case Title:	Read Memberships For a Person
Use case Local ID:	Membership01/05
Brief Description:	A system agent (the Sync Agent) that has an interest in information about people and groups, requests another system agent (the Reference Agent) to provide a list of all the memberships associated with a particular Person. For example, to allow the Sync Agent to identify all the courses upon which a Person is enrolled.
Level:	Sub-function

Use Case Title:	Read Memberships For a Person
Actors:	Primary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Source Agent in some instances and in others a Sync Agent depending on the situation.)
	Secondary: Source Agent – generalized from SRS, MIS, HRMS and TMS. The Source Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	The Person is already known to the Reference Agent and identifying information (in the form of a SourcedId) has been previously communicated to the Sync Agent.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to read membership data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent receives the data for the Memberships it requested.
Triggers:	A user or process requests information about members of a specific Group on the Sync Agent.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system requests Membership information on the Sync Agent about a specific Person object. 2. The Sync Agent sends a request to the Reference Agent to return the set of Membership objects associated with the Person, using a SourcedId to specify the Person. 3. The Reference Agent processes the request and attempts to access and return the object(s). 4. The Reference Agent returns the set of Membership objects to the Sync Agent or the appropriate exception code.
Alternative Flows:	None.
Postconditions:	The Sync Agent has requested that the Source Agent provides data for a set of Memberships associated with a specified group, and has received a response in return.

Use Case Title:	Read Memberships For a Person
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Reference Agent could not locate a Person with the SourcedId provided —The Person has no associated Membership records —Other business rule violations have occurred which prevent object retrieval; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Sync Agent is not permitted by the Reference Agent to request Membership objects) —Encryption error (the Agents could not decrypt the request) —Signing error (the Agents could not guarantee the request had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	This Use Case is related to the following use cases: <ul style="list-style-type: none"> • Read Membership; • Read Memberships for Group.
Notes:	None.

4.6 Reading a Group's Membership

Use Case Title:	Read Memberships For Group
Use case Local ID:	Membership01/06
Brief Description:	A system agent (the Sync Agent) that has an interest in information about people and groups, requests another system agent (the Reference Agent) to provide a list of all the memberships associated with a particular Group. For example, to allow the Sync Agent to identify people and sub-groups for a specific course.
Level:	Sub-function
Actors:	Primary: Sync Agent – generalized from LMS, VLE, LMS and Portal. A Sync Agent is primarily a consumer of information on people and groups that is collected by other systems. (A system may sometimes play the role of a Source Agent in some instances and in others a Sync Agent depending on the situation.)
	Secondary: Source Agent – generalized from SRS, MIS, HRMS and TMS. The Reference Agent is a system of record for information about people and groups, and is interested in keeping other systems that depend upon that information (Sync Agents) synchronized.
Stakeholders & Interest:	Stakeholder: Administrative personnel Interest: Need to ensure that information held on people's membership of groups is consistent across applications, and minimize the need to manually intervene to ensure data is correctly held.
	Stakeholder: Library management personnel Interest: Need their system to hold up-to-date information about memberships of courses held in the MIS/SRS so that resource use is correctly managed.

Use Case Title:	Read Memberships For Group
	Stakeholder: Tutors Interest: Need the learning delivery systems (LMS/VLE) and resource managers (Library) to have the correct people allocated to their courses in the correct roles, and to remain up-to-date when students transfer courses or modules.
	Stakeholder: Student Interest: Need to have administrative systems stay up to date with their status in the organization regarding membership of courses, modules, etc., and to be able to use facilities such as LMS and Library as appropriate.
Preconditions:	The Group is already known to the Reference Agent and identifying information (in the form of a SourcedId) has been previously communicated to the Sync Agent.
Minimal Guarantees:	The Sync Agent and Reference Agent are able to contact each other and the Sync Agent is able to read membership data in the absence of exceptions or in the event of an exception, provide the Reference Agent with the reason for the exception.
Success Guarantees:	The Sync Agent receives the data for the Memberships it requested.
Triggers:	A user or process requests information about members of a specific Group on the Sync Agent.
Basic Flow of Events:	<ol style="list-style-type: none"> 1. An actor such as a member of administrative personnel or another system requests Membership information on the Sync Agent about a specific Group object. 2. The Sync Agent sends a request to the Reference Agent to return the set of Membership objects associated with the Group, using a SourcedId to specify the Group. 3. The Reference Agent processes the request and attempts to access and return the object(s). 4. The Reference Agent returns the set of Membership objects to the Sync Agent or returns an exception code.
Alternative Flows:	None.
Postconditions:	The Sync Agent has requested that the Reference Agent provides data for a set of Memberships associated with a specified group, and has received a response in return.
Exception Handling:	<ul style="list-style-type: none"> • Business-Rule Type Errors <ul style="list-style-type: none"> —The Reference Agent could not locate a Group with the SourcedId provided —The Group has no associated Membership records —Other business rule violations have occurred which prevent object retrieval; • Infrastructure Errors <ul style="list-style-type: none"> —Authentication error (the identity of the Agents cannot be established) —Authorization error (the Sync Agent is not permitted by the Reference Agent to request Membership objects) <ul style="list-style-type: none"> —Encryption error (the Agents could not decrypt the request) —Signing error (the Agents could not guarantee the request had not been changed between transmission and receipt) —Internal error (database connection for example) —Communication error (the message itself could not be correctly interpreted).
Extensions Points:	This Use Case is related to the following use cases: <ul style="list-style-type: none"> • Read Membership; • Read Memberships for Person.
Notes:	None.

About This Document

Title	IMS Enterprise Services Core Use Case Description
Editor	Colin Smythe (IMS)
Team Co-Lead	Chris Vento (WebCT Inc.)
Version	1.0
Version Date	11 June 2004
Status	Final Specification
Summary	This document presents the IMS Enterprise Services Core Use Case Description document. The original Enterprise specification was based upon the description of the data model for the information to be exchanged between communicating enterprise systems. The Enterprise Services specification extends this work by adding a series of behavioral models that define how the data models are to be manipulated. The material in this document describes the core use cases used to generate the Enterprise Services Information Model (not all of the use cases are implemented in v1.0). This version supersedes the IMS Enterprise v1.1 specifications.
Revision Information	11 June 2004
Purpose	This document has been approved by the IMS Technical Board and is made available for adoption.
Document Location	http://www.msglobal.org/es/esv1p0/imses_usecasev1p0.html

To register any comments or questions about this specification please visit:
<http://www.msglobal.org/developers/ims/imsforum/categories.cfm?catid=20>

List of Contributors

The following individuals contributed to the development of this document:

Name	Organization	Name	Organization
Scott Baker	Oracle Inc.	Les Smith	SCT Inc.
Fred Beshears	UC Berkeley, USA	Colin Smythe	Dunelm Services Ltd.
Kerry Blinco	IMS Australia	Scott Thorne	MIT, USA
Mark Bolenbach	SCT Inc.	Chris Vento	WebCT Inc.
Chris Etesse	Blackboard Inc.	Kimberley Voltero	WebCT Inc.
John Hallet	WebCT Inc.	Scott Wilson	JISC (CETIS), UK
Cathy Schroeder	Microsoft Inc.	Nathaniel Zinn	Blackboard Inc.

Revision History

Version No.	Release Date	Comments
Base Document 1.0	21 July 2003	The final form of the Core Use Case Description Base Document. This outline document is submitted to the IMS TB for voting approval.
Public Draft 1.0	12 January 2004	The final form of the Core Use Case Description Public Draft. This document is made available for public review and comment.
Final Specification 1.0	11 June 2004	This is the formal Final Release of the IMS Enterprise Services specification.

Index

- A**
 - Abstract Framework 3, 4, 5
 - API 4
 - Attributes
 - Membership
 - membership 4, 19, 25, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39
 - Result
 - result 3
 - Role
 - status 19, 21, 23, 25, 29, 31, 33, 35, 37, 39
 - StatusInfo
 - description 3, 40
 - Values
 - list 10, 23, 34, 36, 38
- B**
 - Binding technologies
 - WSDL 3, 4, 5
- C**
 - Classes
 - Common
 - Identifier 12
 - Group 3, 4, 5, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 34, 37, 38, 39
 - Description 3, 5, 6, 7, 8, 11, 12, 14, 15, 17, 19, 20, 22, 24, 26, 29, 31, 32, 35, 36, 38, 40, 41
- E**
 - Enterprise Service 3, 4, 5, 40
- G**
 - Group Management Service 3, 4, 5
- M**
 - Membership Management Service 3, 4
- P**
 - Person Management Service 3, 4, 5
- S**
 - Schools Interoperability Framework 4
- U**
 - Use-case
 - Group
 - Creating 19
 - Deleting 24, 26
 - Deleting a Group Relationship 26
 - Reading 20
 - Updating 22
 - Membership
 - Creating 29
 - Deleting 35
 - Reading 31
 - Updating 32
 - Person
 - Creating 6
 - Deleting 11
 - Querying Created Persons 14
 - Querying Deleted Persons 17
 - Querying Updated Persons 15
 - Reading 7
 - Updating 8
- W**
 - WDSL 3, 4, 5

IMS Global Learning Consortium, Inc. ("IMS") is publishing the information contained in this IMS Enterprise Services Core Use Case Descriptions ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.

IMS 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 would appreciate receiving your comments and suggestions.

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

Please refer to Document Name: IMS Enterprise Services Core Use Case Descriptions

Date: 11 June 2004