IMS Enterprise Information Model

Version 1.01

About This Document

Title	IMS Enterprise Information Model
Authors	Geoff Collier, Wayne Veres
Version	1.01
Version Date	December 21, 1999
Status	Final
Summary	This document describes the IMS Enterprise Information Model which is used to support process interoperability between Learning Management Systems and Enterprise systems such as corporate human resources management, student administration, and library management systems.
Revision Information	Last revised December 21, 1999
Document Location	http://www.imsproject.org/enterprise/eninfo03.pdf

Table of Contents

Introduction

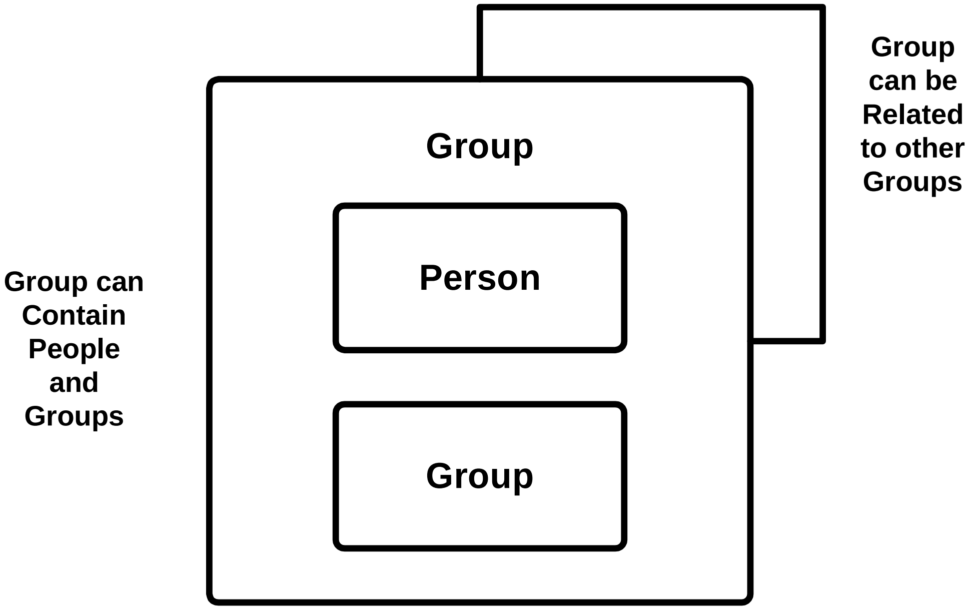
The IMS Enterprise Information Model describes data structures that are used to provide interoperability of Internet-based Instructional Management systems with other Enterprise systems used to support the operations of an organization.

The objective of the IMS Enterprise Information Model is to define a standardized set of structures that can be used to exchange data between different systems. These structures provide the basis for standardized data bindings that allow software developers and implementers to create Instructional Management processes that interoperate across systems developed independently by various software developers. The major classes of Enterprise applications supported by this model are Training Administration, Student Administration, Library Management, and Human Resource systems.

Note: The scope of the IMS Enterprise specification is focused on defining interoperability between systems residing within the same enterprise or organization. The documents comprising the IMS Enterprise specification are not targeted at solving the issues of data integrity, communication, overall security, and others that are inherent when investigating cross-enterprise data exchange.

Overall Data Model

The following diagram provides a conceptual overview of the IMS Enterprise Interoperability data model.



DATA OBJECTS:

This model is supported through the use of three data objects, described briefly below:

Person – This data object contains elements describing an individual of interest to the Learning Management environment.

Group – This object contains elements describing a group of interest to the Learning Management environment. There are many types of groups that may be shared between systems. The most common is a Course Instance, but they may also include Training Programs, Academic Programs, Course sub-groups, clubs, etc. A group can also have any number of relationships with other groups.

Group Membership – This data object contains elements describing the membership of a person or group in a group. Group members may be instructors, learners, content developers, members, managers, mentors, or administrators.

A NOTE ON REFERENTIAL INTEGRITY:

In the information model shown above, there is an implied referential integrity between data objects. For example, defining a Group Membership instance first requires the existence of the Group, and of the Person or Group that is a member of the Group

Conceptual Description of the Data Objects

The tables in this section provide a conceptual, informative description of the elements contained in the data objects.

Description of the columns in these tables:

No: The number of the data element. An element may be composed of sub-elements. The numbering scheme reflects these relationships.

Name: The descriptive name of the element.

Explanation: A brief functional description of the element.

Required: Indicates if the element is required.

- **M** = Mandatory Element that must be included in the data object, if the element at the higher level is included. For example, in the Person object, Name is a mandatory element of the Person Object, and FN is a mandatory element of Name. Another example, TelNum is a mandatory element of Tel, although Tel is an optional element of the Person Object.

- **C** = Conditional Element. Existence is dependent on values of other Elements.

- **O** = Optional Element

Multi: Multiplicity of the element.

- Blank = single instance

- Number = maximum number of times the element is repeatable

- **n** = Multiple occurrences allowed, no limit

- Repeatability of an element implies that all sub-elements repeat with the element

Domain: A description of the set of valid values for the element.

- Coding schemes use numeric codes rather than alphabetic representations (e.g. “1” for male rather than “M”) in order to make them language independent.

- If a set of values is defined in the Domain, then the domain is closed. The list of values cannot be extended to include values not defined in the specification. If there is a need for values not included in the domain set of values, then the extension should be done by defining a new element under the Extension element that is part of each data object definition.

Type: A description of formatting rules for the data element. Type includes the maximum length of the element.

- ID = element used to uniquely identify an object

- Code = element value from a list of codes

- Descr = descriptive element, human language

- Flag = binary flag

- The international character set specified by ISO 10646 will be used for all fields.

Notes: Additional descriptive information about the element.

Example: Illustration of legitimate values for this element.

A NOTE ON EXTENSIBILITY:

In this specification, an EXTENSION element is defined for each data object. Extensions are to be implemented in structures that are sub-elements of this element. This applies to all extensions, including extensions to valid codes for elements that have a domain set of valid values defined as part of the specification. Examples of valid extensions are provided in the IMS Enterprise Best Practices and Implementation Guide.

Properties Data Object

When a source system generates one or more data objects, it must include some basic packaging and control data that the target system(s) use to determine the source, timing, and type of event that has caused the generation of a data package. This information is used to manage the interchange of data between systems.

No.	Name	Explanation	Reqd	Mult	Domain	Type	Note	Example
1.1	**DataSource**	An identifier for the source system	M			ID String 256	-Allows the target system(s) to identify the system that generated the data objects - Value must be unique for each source	
1.2	**Target**	An identifier for a target system	O	n		ID String 256	- If the data objects are intended for one or more specific target systems, this element identifies those systems - Must use the same system naming scheme as is used for data source	
1.3	**Type**	Describes the type of event that caused the source system to generate the data objects	O			Code String 32	- A standard set of codes must be agreed upon for any specific implementation	Examples: - Initial Group Creation - Group maintenance - Full group membership - Membership changes - Final grades
1.4	**Datetime**	Date and time the data objects were generated by the data source	M			Date time in ISO8601 standard format		

1.5	**Lang**	Default language of the text information provided in the attached objects	O		Language codes defined in ISO 639	Code String 128		
1.6	**Extension**	Acts as the high level element for any extensions to the data object	O				All extensions are to be implemented as sub-elements under this element	

Person Data Object

When a person data object is passed, the target system will update its files with the data in the object. If it is a new person, they will be added to the system. If a person's data has changed, the new data should be used to update the target system. If a person is flagged for deletion, it is up to the target system to determine what action to take with that deletion information.

The target system needs to be capable of storing the source system's "Sourced ID", which is used to uniquely identify a person within the implementation environment. This is required to support future updates of person information from the source system.

No.	**Name**	**Explanation**	**Reqd**	**Mult**	**Domain**	**Type**	**Note**	**Example**
2.1	**SourcedID**	The ID of a person as defined by the source system	M				In order to be unique, a Person ID must include both an identifier of the system where the Person object was created, and a unique identifier within that system.	
2.1.1	**Source**	Identifier of the organization or system that assigned the ID	M			ID String 32		
2.1.2	**ID**	Permanently unique identifier for a person, as defined by the system where the person object was created	M			ID String 256		

2.2	**RecStatus**	Record Status - Indicates that the person has been added, updated, or deleted in the source system	O		1 = Add 2 = Update 3 = Delete	Code Numeric 1	In an event driven interface, this flag can be passed from the source to the target system to indicate that the person has been added, updated, or deleted.	
2.3	**UserID**	Person’s user ID to access the Learning Management environment	O			Code String 256	Any implementation of this specification must ensure the security of this, and all, data.	
2.4	**Name**	The name of the person	M				Note that the name parts specified below are to support interoperability, not to maintain a full record of a person’s names.	
2.4.1	**FN**	Formatted name	M			Descr String 256		- Robert Lawdon Jones III - Shirley R. Smith Jr. - Ling Pao - Basten Holter
2.4.2	**Sort**	Name “parsed” and re-ordered so it will sort appropriately on an alphabetized report or online panel (This name is never displayed, it is only used to sort.)	O			Descr String 256	“Parsing” schemes will be vendor specific. The examples at right use a scheme that concatenates the first five letters of family name followed by the first five letters of Given name.	“HOLTEBAS ” rather than Basten Holter “JONESROBER” rather than Robert Jones “LING PAO” rather than Ling Pao

2.4.3	**Nickname**	Full name formatted in the way that the person prefers to be addressed	O			Descr String 256		Bob Jones, rather than Robert Lawdon Jones III.
2.4.4	**N**	Name with all parts distinguished	O					
2.4.4.1	**Family**	Note that this is the Family name, not the Last name (The order of name parts varies by culture.)	O			Descr String 256		Jones
2.4.4.2	**Given**	Given name, not necessarily first name (The order of name parts varies by culture.)	O			Descr String 256		Robert
2.4.4.3	**Other**	Other name parts	O	n		Descr String 256		Lawdon
2.4.4.4	**Prefix**		O			Descr String 32		Mr, Mrs, Ms, Mssr, Dr, etc
2.4.4.5	**Suffix**		O			Descr String 32		Jr, III, Sr, etc.

2.5	**Demographics**	Demographic information about the person	O				Minimal demographic information is specified. This is useful for the confirmation of identity. Other data elements such as citizenship, ethnicity, and place of birth have been considered for the standard specification, but no specific interoperability need has been found for these elements as yet.	
2.5.1	**Gender**	Gender of the person	O		0 = unknown 1 = female 2 = male	Code String 1		
2.5.2	**BDay**	Date the person was born	O		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		
2.6	**Email**	Email address used to contact a person	O			Descr String 256	No repeatability? supported in specification, because no specific interoperability need has been identified for more than one occurrence of Email Address.	
2.7	**Tel**	Telephone number used to contact a person	O	2				

2.7.1	**TelType**	Indicates what type of phone number is being specified.	O		1 = Voice 2 = Fax	Code String 8		
2.7.2	**TelNum**	Telephone number	M			Code String 32	International format should be used. An appropriate, widely accepted standard has not been agreed on yet. People implementing the spec will have to decide on their own format.	One example of a format is:+ccc (aaa) nnnnn nnnnn . +c = Country Code. (a) = Area Code . n = lbersocal num blanks are allowed
2.8	**Adr**	Address used to deliver physical objects to a person.	O				Only one address is supported in the specification because no specific interoperability need has been identified for more than one Address.	
2.8.1	**POBox**	Post Office Box	O			Descr String 32		
2.8.2	**ExtAdd**	Extra address data	O			Descr String 128	Any “non street” components of the address, suite number, company name, care of, etc,	
2.8.3	**Street**	Street address	O	3		Descr String 128	Ordered list--should be used in the order presented	

2.8.4	**Locality**	Locality	O			Descr String 64	City is one example of Locality	
2.8.5	**Region**	Region	O			Descr String 64	State and Province are examples of Region	
2.8.6	**PCode**	Postal Code	O			Descr String 32	Format of postal code varies by country	
2.8.7	**Country**	Country	O		Codes specified in ISO 3166	Descr String 64		
2.9	**Photo**	Reference to an external location containing a photo of the person	O					
2.9.1	**ExtRef**	The reference to an external location	M			Descr String 1024	Could be a URL-- Web standards for location references are not finalized. When they are, this specification will incorporate them.	
2.9.2	**ImgTyp**	The type of image referred to	O			Descr String 32	Should contain the IANA registered image type	bmp jpg gif

2.10	**DataSource**	An identifier of the source system for the person object.	O			ID String 256	Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects.	
2.11	**Extension**	Acts as the high level element for any extensions to the data object	O				All extensions are to be implemented as sub-elements under this element.	

Group Data Object

When a group data object is passed, the target system will update its files with the data in the object. If it is a new group, it will be added to the system. If a group's data has changed, the new data should be used to update the target system. If a group is flagged for deletion, it is up to the target system to determine what action to take with that deletion information.

The target system needs to be capable of storing the source system's "Sourced ID", which is used to uniquely identify a group within the implementation environment. This is required to support future updates of group information from the source system.

It is possible to have a viable interface without automatically exchanging Group data, but this means that one or the other of the systems involved in an interface must first store the other system's Group Identifier in order to support the passing of Group membership data.

A NOTE ON META-DATA LEARNING GROUP DESCRIPTIONS: The IMS Meta-data Specification is designed to allow systems to interchange descriptive information about a learning object, including learning groups such as course instances for which this IMS Enterprise Specification also defines a data interchange object. The difference is that the Enterprise Group Data Object described below is designed to pass data related to the operation and management of a learning group, whereas the Meta-data Object is designed to describe the content of a learning object. Therefore, if there is a need to pass more extensive descriptive information about a learning group between systems than is allowed by this Enterprise Specification, then the Meta-data Specification should be used as the guide for passing the descriptive information.

No.	Name	Explanation	Reqd	Mult	Domain	Type	Note	Example
3.1	**SourcedID**	Unique group identifier	M				In order to be unique, a Group ID must include both an identifier of the system where the Group object was created, and a unique identifier within that system.	
3.1.1	**Source**	Identifier of the organization or system that created the group object	M			ID String 32		

3.1.2	**ID**	Permanently unique identifier for a group, as defined by the system where the group object was created	M			ID String 256		The standard supports any type of ID, from a term id and sequential number typically assigned, to a course instance in a student administration system, to the URL address of a course instance in an Internet Learning Management System.
3.2	**RecStatus**	Record Status - Indicates that the group has been added, updated or deleted from in source system	O		1 = Add 2 = Update 3 = Delete	Flag Numeric 1	In an event driven interface, this flag can be passed from the source to the target system to indicate that the group has been added, updated, or deleted.	
3.3	**GroupType**	Defines what type of group this is	O	n			This element provides a structure that allows a Group to be categorized into one or more coding schemes, with any number of levels supported within each scheme.	
3.3.1	**Scheme**	Group Type coding scheme	O				Identifies which Group categorization scheme is being used. This could be a proprietary vendor taxonomy, a national subject area taxonomy, etc.	
3.3.2	**TypeValue**		M	n			Repeats to allow more than one level of code to be stored.	

3.3.2.1	**Level**	Group Type code level	M			Numeric String 2	Level 1 is the highest level, level 2 provides a further refinement of the level 1 category, etc.	
3.3.2.2	**Value**	Group Type code value	M			Descr String 256	The value at this level	An example of the Level / Value interaction might be: Lvl 1– Instruction Lvl 2– Discussion Group Lvl 3- Web enabled
3.4	**Description**	Description / Name of the Group	M					
3.4.1	**Short**	Intended to be displayed on screen on less than one line.	M			Descr String 60		Usually something brief such as “ENGLISH 101A SECTION 4”.
3.4.2	**Long**	Longer descriptive name for the group	O			Descr String 256		“English 101A – Great Authors of the 19th and 20th Century”
3.4.3	**Full**	A longer description of the group	O			Descr String 2048		For example, the “catalog” description of a course .
3.5	**Org**	The organization administering or “sponsoring” the Group	O					For example, Cal State San Marcos would be the administrator of a course section offered on their campus.

3.5.1	**OrgName**	The name of the organization	M			Descr String 256		“Cal State San Marcos”
3.5.2	**OrgUnit**	Name of sponsoring or administering unit within the organization	O	n		Descr String 256	One or more departments or units can sponsor the group.	“0158 – Math Department”
3.5.3	**Type**	Used to distinguish general categories of organization	O			Code String 32		“Academic Unit” “HR Department”
3.5.4	**ID**	ID of the organization	O			ID String 256	If there is a code for the organization, it can be specified separately in this field.	
3.6	**TimeFrame**	Time frame when the group is active	O					
3.6.1	**Begin**	Defines when a group is intended to be available for participation	O					
3.6.1.1	**Date**	Date of availability	M		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		
3.6.1.2	**Restrict**	Do not allow learner participation until the begin date	C		0 = no 1 = yes	Flag Numeric 1 fixed	This field can only exist if the begin date is provided.	
3.6.2	**End**	Defines when group participation is intended to end	O					

3.6.2.1	**Date**	Date participation is intended to end	M		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		
3.6.2.2	**Restrict**	Do not allow learner participation after the end date	C		0 = no 1 = yes	Flag Numeric 1 fixed	This field can only exist if the end date is provided.	
3.6.3	**AdminPeriod**	Short descriptive name in human readable form for the administrative or academic period within which the group exists	O			Code String 32		- 1999 - Fall 1999 - Summer 1999 Session 2
3.7	**EnrollControl**		O					
3.7.1	**EnrollAccept**	Indicates if the Group is accepting enrollments	O		0 = no 1 = yes	Flag Numeric 1 fixed	There can be different reasons for a group being closed. It may be full; it may be cancelled.	
3.7.2	**EnrollAllowed**	Determines if target system can enroll people	O		0 = no 1 = yes	Flag Numeric 1 fixed	If No, then only the source system can enroll people.	
3.8	**Email**	Email address used to contact all members of a group	O			ID String 256	Email distribution address for the group.	

3.9	**URL**	URL for a group	O			Descr String 256	For a course, this would be the course home page.	
3.10	**Relationship**	If the group is related to another group, then this Element can be used to describe that relationship	O	n			The relationship described is the relationship of the group to the Group defined in the Group object. For example, if the Label element says “Subgroup” and the Relation element is “2” (child), then the group in this element is a subgroup, and a child of the group defined in the group object. Note that this relationship segment should not be used to store “membership” in other groups. The Group membership construct is used for that type of role-based membership.	
3.10.1	**SourcedID**	Unique group identifier	M				ID of the other group, with which we are establishing a relationship.	
3.10.1.1	**Source**	Identifier of the organization or system that created the group object	M			ID String 32		
3.10.1.2	**ID**	Permanently unique identifier for a group, as defined by the system where the group object was created	M			ID String 256		

3.10.2	**Label**	Describes the nature of the relationship between this group and the related group.	M			Code String 32		“Course Subgroup” “Cross Listed Course Section”
3.10.3	**Relation**	Defines the “hierarchical” nature of the relationship, if there is such a hierarchy	O		1=parent 2=child 3 =also known as	Code String 8	Relation should be read as “Group is Relation of the Relationship group”. That is, if the group object is “Marketing Division”, the relationship group is “Northwest Region”, and the relation is “1”, then the logical relationship is “Marketing Division is the parent of Northwest Region”. Code “3” (also known as) is used to indicate that two groups are really the same group. An example of this might be cross listed course sections.	
3.11	**DataSource**	An identifier of the source system for the group object	O			ID String 256	Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects.	
3.12	**Extension**	Acts as the high level element for any extensions to the data object	O				All extensions are to be implemented as sub-elements under this element.	

Membership Data Object

This object is primarily intended to pass group membership information from systems that manage various types of group enrollment processes to the systems that provide Learning Management services to those group members. The enrollment processes referred to here include a wide range of specific functionality, including but not limited to, examples such as:

- enrollment in a course,
- employment in a particular company division, or
- membership in a club.

In addition, it allows final result data to be exchanged (typically final grade or a completion indicator) so it can also be used to pass the final result data back from the learning system to the enterprise system.

When a membership data object is passed, the target system will update its files with the data in the object. If it is a new membership , it will be added to the system. If membership records data has changed, the new data should be used to update the target system. . If a membership is flagged for deletion, it is up to the target system to determine what action to take with that deletion information.

No.	Name	Explanation	Reqd	Mult	Domain	Type	Note	Example
4.1	**SourcedID**	Unique group identifier	M				.	
4.1.1	**Source**	Identifier of the organization or system that created the group object	M			ID String 32		
4.1.2	**ID**	Permanently unique identifier for a group, as defined by the system where the group object was created	M			ID String 256		
4.2	**Member**	Group Member	M	n				

4.2.1	**SourcedID**	The ID of a person or group as defined by the source system	M				In order to be unique, a Person ID must include both an identifier of the system where the Person object was created, and a unique identifier within that system.	
4.2.1.1	**Source**	Identifier of the organization or system that assigned the ID	M			ID String 32		
4.2.1.2	**ID**	Permanently unique identifier for a person or group, as defined by the system where the person or group object was created	M			ID String 256		
4.2.2	**IDType**	Indicates if the member is a person, or another group.	M		1 = Person 2 = Group	Flag Numeric 1	A member can either be a person or another group.	
4.2.3	**Role**		M	n			- A member can have multiple roles in a group (for example Learner and Instructor). These would be reflected in separate occurrences of the Role element.	

4.2.3.1	**RoleType**	The member’s function within a group	M		01=Learner 02=Instructor 03=Content Developer 04=Member 05=Manager 06=Mentor 07=Administrator	Code String 32		
4.2.3.2	**SubRole**	Further qualifies a member’s role in the group	O			Code String 32		For an Instructor RoleType, examples are: Primary Instructor Teaching Assistant Tutor

4.2.3.3	**Status**	Indicates if a member is active or inactive in the group	M		0 = Inactive 1 = Active	Flag Numeric 1 fixed	- This allows the source system to specifically tell the target system that a member is now active or inactive.- Another view is that the absence of a membership record when membership data is passed implies inactivity, and the existence of a record implies active membership. This will logically work for a “snapshot” interface where all members are passed every time objects are passed from one system to another, but it will not support an interface where individual membership records are passed.	
4.2.3.4	**RecStatus**	Record Status - Indicates that the group membership has been added, updated, or deleted from in source system	O		0 = Add 2 = Update 3 = Delete	Flag Numeric 1	In an event driven interface, this flag can be passed from the source to the target system to indicate that the group membership has been added, updated, or deleted.	
4.2.3.5	**UserID**	Person’s user ID to access the group for this role	O			Code String 256	Any implementation of this specification must ensure the security of this, and all, data.	

4.2.3.6	**Comments**	Description of the current status	O			Descr String 2048	May be used to describe why a member’s status changed, or simply to record more detail about a member’s status in the group.	
4.2.3.7	**Date**	Date the current membership status was established	O		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		
4.2.3.8	**Timeframe**	Time frame of membership in a group	O					
4.2.3.8.1	**Begin**	Defines when a group is intended to be available for participation for this member.	O					
4.2.3.8.1.1	**Date**	Date of availability	M		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		
4.2.3.8.1.2	**Restrict**	Do not allow member participation until the begin date	C		0 = no 1 = yes	Flag Numeric 1 fixed	This field can only exist if the begin date is provided.	
4.2.3.8.2	**End**	Defines when group participation is intended to end for this member	O					
4.2.3.8.2.1	**Date**	Date participation is intended to end	M		0001-01-01 to 9999-12-31	Date in ISO8601 standard format		

4.2.3.8.2.2	**Restrict**	Do not allow member participation after the end date	C		0 = no 1 = yes	Flag Numeric 1 fixed	This field can only exist if the end date is provided.	
4.2.3.9	**Final result**	Final result codes and value	O				The specification allows for the passing of a group member's final result mode and all valid result values from the source system to the target system. It is provided at the Group member level because it can vary for different members of a Group (e.g. – one learner could be on a graded basis, another pass/fail, and another auditing, with no valid result codes). The specification also allows for the passing of a group member's final result, both a result code and result description.	
4.2.3.9.1	**Mode**	Short descriptive name for final result grading mode	O			Code String 32		"Letter Grade" "Pass/Fail" "Percentage" "Attendance"
4.2.3.9.2	**Values**	Valid result values.	O			Code String 2048	Used to tell the target system what final result values are valid to assign to a learner	

4.2.3.9.2.1	**ValueType**	Indicates if the values are a list of specific codes, or a numeric range	M		0 = List 1 = Range	Flag Numeric 1 Fixed	Tells the system to use either the list or the Min / Max values.	
4.2.3.9.2.2	**List**	A specific result value	C	n		Code String 32	The list contains the valid grades if Value Type = 0	
4.2.3.9.2.3	**Min**	Minimum numeric value allowed for a result	C			Code Numeric 8 to 4 decimals	Used if Value type = 1	
4.2.3.9.2.4	**Max**	Maximum numeric value allowed for a result	C			Code Numeric 8 to 4 decimals	Used if Value type = 1	
4.2.3.9.3	**Result**	Value of final result assigned to the member for participation in the group.	O			Code String 32	Ideally, this would be one of the values from the Result values list.	- A+ - 85% - Completed - Attended
4.2.3.9.4	**Comments**	Comments about final result	O			Descr String 2048	Can be used for a descriptive evaluation of a learner’s participation in a group	
4.2.3.10	**Email**	Email address used to contact a member for information related to a specific group membership	O			Descr String 256	Used if the member has a different Email address for a specific group.	

4.2.3.11	**DataSource**	An identifier of the source system for the membership object	O			ID String 256	Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects. Note that this is specified at the role level.	For example, Student roles could come from one source system, and Learner roles could come from another source system, and both be transmitted in the same file of data objects.
4.2.3.12	**Extension**	Acts as the high level element for any extensions to the data object	O				All extensions are to be implemented as sub-elements under this element.	

Addendum

Changes from Version 1.0 of the Enterprise Systems Information Model include the following:

Date:	December 14, 1999
Document	IMS Enterprise Information Model v 1.0
Submitter	Wayne Veres, Geoff Collier

Type of Problem	Technical Error
Date:	December 14, 1999
Problem Description	Inconsistency between the Information Model and the XML Binding Specification
References in Document	Enterprise Information Model p. 6 – Language element
Solution Proposed by the Submitter	1) Element called **Language** should be renamed to **Lang** to be consistent with Enterprise and Metadata DTDs.

Type of Problem	Clarification
Date:	December 14, 1999
Problem Description	Group relation description does not make it completely clear how the relation ties the two groups together.
References in Document	Enterprise Information Model p. 20 – Relation description
Solution Proposed by the Submitter	2) Insert the following paragraph in front of the existing comment. Relation should be read as "Group is Relation of the Relationship group". That is, if the group object is "Marketing Division", the relationship group is "Northwest Region", and the relation is "1", then the logical relationship is "Marketing Division is the parent of Northwest Region".

Type of Problem	Addition
Date:	December 14, 1999
Problem Description	DataSource is needed in the Person, Group and Membership objects, not just in the Properties object. This allows a single file to contain objects from more than one source system, and allows the target system to track the source of the objects for future reference back.
References in Document	Enterprise Information Model – p. 13, 20, 28
Solution Proposed by the Submitter	3) Make the following additions and changes. **Page 13, Person Object:** Renumber the "Extension" element as 2.11 Insert the "DataSource" element before Extension, as follows: No - "2.10" Name – "DataSource" Explanation – "An identifier of the source system for the person object" Reqd – "O" Mult – blank Domain – blank

	Type – “ID String 256” Note – “Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects.” Example – blank **Page 20, Group Object:** Renumber the “Extension” element as 3.12 Insert the “DataSource” element before Extension, as follows: No - “3.11” Name – “DataSource” Explanation – “An identifier of the source system for the group object” Reqd – “O” Mult – blank Domain – blank Type – “ID String 256” Note – “Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects.” Example – blank **Page 28, Group Membership Object:** Renumber the “Extension” element as 4.2.3.12 Insert the “DataSource” element before Extension, as follows: No - “4.2.3.11” Name – “DataSource” Explanation – “An identifier of the source system for the membership object.” Reqd – “O” Mult – blank Domain – blank Type – “ID String 256” Note – “Used to identify the source system for the data object. Allows more than one source system to provide objects in the same file of data objects. Note that this is specified at the role level.” Example – “For example, Student roles could come from one source system, and Learner roles could come from another source system, and both be transmitted in the same file of data objects.”