



IMS Content Packaging Information Model

Version 1.1.2 Final Specification

**Copyright © 2001 by IMS Global Learning Consortium, Inc.
All Rights Reserved.**

The IMS Logo is a trademark of IMS Global Learning Consortium, Inc.
Document Name: IMS Content Packaging Information Model Revision: 8 August 2001

Table of Contents

TABLE OF CONTENTS	2
1. INTRODUCTION	3
1.1 OVERVIEW	3
1.2 SCOPE & CONTEXT.....	3
1.3 STRUCTURE OF THIS DOCUMENT	3
1.4 NOMENCLATURE.....	3
2. IMS CONTENT PACKAGING CONCEPTUAL MODEL	4
2.1 KEY ELEMENTS.....	4
2.2 STANDARD NAME FOR THE MANIFEST FILE	5
3. EXTENSIBILITY	6
4. MANIFEST ELEMENTS	7
4.1 (SUB)MANIFESTS	12
APPENDIX A - LIST OF CONTRIBUTORS	14
ABOUT THIS DOCUMENT	15
REVISION HISTORY	16
INDEX	18

1. Introduction

1.1 Overview

The IMS Content Packaging Information Model describes data structures that are used to provide interoperability of Internet based content with content creation tools, learning management systems (LMS), and run time environments.

The objective of the IMS Content Packaging Information Model is to define a standardized set of structures that can be used to exchange content. These structures provide the basis for standardized data bindings that allow software developers and implementers to create instructional materials that interoperate across authoring tools, LMSs, and run time environments that have been developed independently by various software developers.

Note: The scope of the IMS Content Packaging specification is focused on defining interoperability between systems that wish to import, export, aggregate, and disaggregate Packages of content. Future documents comprising the IMS Content specification will address requirements regarding content data models and communication between run time environments and LMSs.

1.2 Scope & Context

This document is the IMS Content Packaging (CP) Information Model specification. As such it will be used as the basis for the production of the following documents:

- IMS Content Packaging XML Binding v1.1.2
- IMS Content Packaging Best Practice Guide v1.1.2

1.3 Structure of this Document

The structure of the rest of this document is:

- | | |
|---|--|
| 2. IMS Content Packaging Conceptual model | The underlying usage, processing control and data structures comprising Content Packaging. |
| 3. Extensibility | The ways in which proprietary extensions are supported through this specification. |
| 4. Manifest Elements | The detailed description of the Manifest elements in terms of their properties and attributes. |
| Appendix A - Contributors | A listing of individuals who contributed to the development of this document. |

1.4 Nomenclature

CDATA	Character Data
CPI	Content & Packaging Interchange
DTD	Document Type Definition
PCDATA	Parsed Character Date
W3C	World Wide Web Consortium
XML	Extensible Mark-up Language

2. IMS Content Packaging Conceptual Model

In Figure 2.1 is a conceptual diagram that illustrates the components of the IMS Content Packaging Information Model. As indicated in the IMS Content Packaging Best Practice Guide, this is part of the larger IMS Content Framework, which forms the basis for this and future specifications.

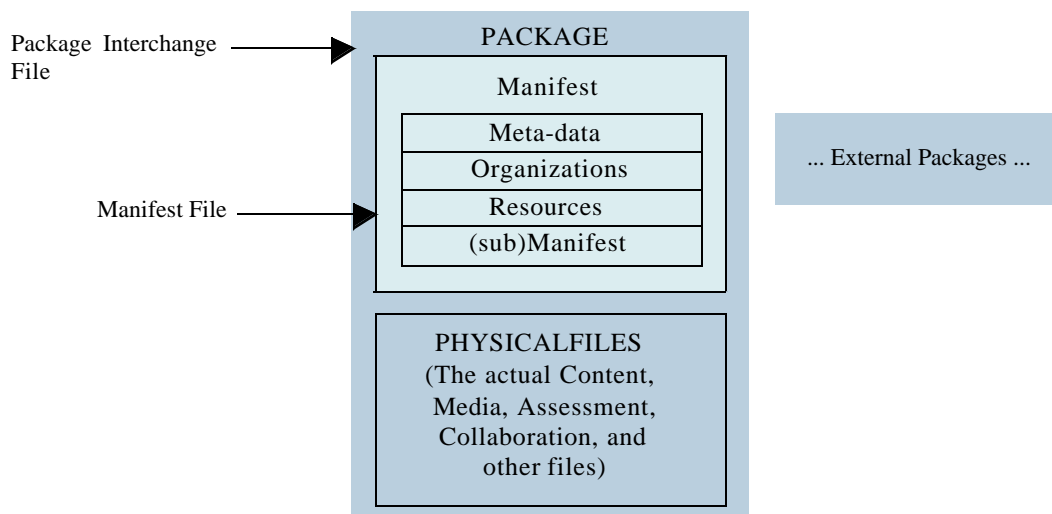


Figure 2.1 IMS Content Packaging scope.

2.1 Key Elements

The IMS Package depicted in Figure 2.1 consists of two major elements: a special XML file describing the content organization and resources in a Package, and the physical files being described by the XML. The special XML file is called the IMS Manifest file, because course content and organization is described in the context of ‘manifests’. Once a Package has been incorporated into a single file for transportation, it is called a Package Interchange File. The relationship of these parts to the content container is described below:

Package Interchange File – a single file, (e.g., .zip, .jar, .cab) which includes a top-level manifest file named “imsmanifest.xml” and all other physical files as identified by the Manifest. A Package Interchange File is a concise Web delivery format, a means of transporting related, structured information. PKZip v2.04g (.zip) is recommended as the default Package Interchange File format.

Package – a logical directory, which includes a specially named XML file, any XML control documents it references (such as a DTD or XSD file), and sub-directories containing the actual physical resources.

- **Top-level Manifest** – a mandatory XML element describing the Package itself. It may also contain optional (sub)Manifests. Each instance of a manifest contains the following sections:
 - **Meta-data section** – an XML element describing a manifest as a whole.
 - **Organizations section** – an XML element describing zero, one, or multiple organizations of the content within a manifest.
 - **Resources section** – an XML element containing references to all of the actual resources and media elements needed for a manifest, including meta-data describing the resources, and references to any external files.
 - **(sub)Manifest** – one or more optional, logically nested manifests.
- **Physical Files** – these are the actual media elements, text files, graphics, and other resources in their various sub-directories as described by the manifest(s).

Package – A Package represents a unit of usable (and reusable) content. This may be part of a course that has instructional relevance outside of a course organization and can be delivered independently, as an entire course or as a collection of courses. Once a Package arrives at its destination to a run time service, such as an LMS vendor, the

Package must allow itself to be aggregated or disaggregated into other Packages. A Package must be able to stand-alone; that is, it must contain all the information needed to use the contents for learning when it has been unpacked.

Packages are not required to be incorporated into a Package Interchange File. A Package may also be distributed on a CD-ROM or other removable media without being compressed into a single file. An IMS Manifest file and any other supporting XML files required by it (DTD, XSD) must be at the root of the distribution medium.

Manifest – A manifest is a description in XML of the resources comprising meaningful instruction. A manifest may also contain zero or more static ways of organizing the instructional resources for presentation.

The scope of manifest is elastic. A manifest can describe part of a course that can stand by itself outside of the context of a course (an instructional object), an entire course, or a collection of courses. The decision is given to content developers to describe their content in the way they want it to be considered for aggregation or disaggregation. The general rule is that a Package always contains a single top-level manifest that may contain one or more (sub)Manifests. The top-level manifest always describes the Package. Any nested (sub)Manifests describe the content at the level to which the (sub)Manifest is scoped, such as a course, instructional object, or other.

For example, if all content comprising a course is so tightly coupled that no part of it may be presented out of the course context, a content developer would want to use a single manifest to describe that course's resources and organization. However, content developers who create "instructional objects" that could be recombined with other instructional objects to create different course presentations would want to describe each instructional object in its own manifest, then aggregate those manifests into a higher-level manifest containing a course organization. Finally, a content developer who wants to move multiple courses in a single Package (a curriculum), would use a top-level manifest to contain each course-level manifest and any instructional object manifests that each course might contain.

Resource – The resources described in the manifest are physical assets such as web pages, media files, text files, assessment objects or other pieces of data in file form. Resources may also include assets that are outside the Package but available through a URL, or collections of resources described by (sub)Manifests. The combination of resources is generally categorized as content. Each resource may be described in a <resource> element within a manifest's XML. This element includes a list of all the assets required to use the resource. The files included in the Package are listed as <file> elements within such <resource> elements.

2.2 Standard Name for the Manifest File

Content distributed according to the IMS Content Packaging specification must contain an IMS Manifest file. To ensure that the IMS Manifest file can always be found within a Package, it has a **pre-defined name** and location:

```
imsmanifest.xml
```

The IMS Manifest file and any of its supporting XML files (DTD, XSD) must be placed at the root of the Package Interchange File or any other packaging image (like a CD-ROM).

In the absence of this file, the package is **not** an IMS Package and cannot be processed. It is required that the name be kept, as above, in all lowercase letters.

3. Extensibility

An important underpinning of the IMS Content Packaging specification is rich support for extensibility. While the base Content Packaging Information Model leverages the rich set of meta-data elements defined in the IMS Meta-Data Specification v1.2, it defines only the basic structures for organization and resources (Web Content). It is expected that implementers of this specification will define new types of resources and organizations to describe and transport rich learning resources, and over time, it may be possible to incorporate widely used extensions into future versions of this specification.

4. Manifest Elements

This section provides a conceptual, informative description of the elements contained in a Manifest. Figure 4.1 illustrates the primary elements of a Manifest.

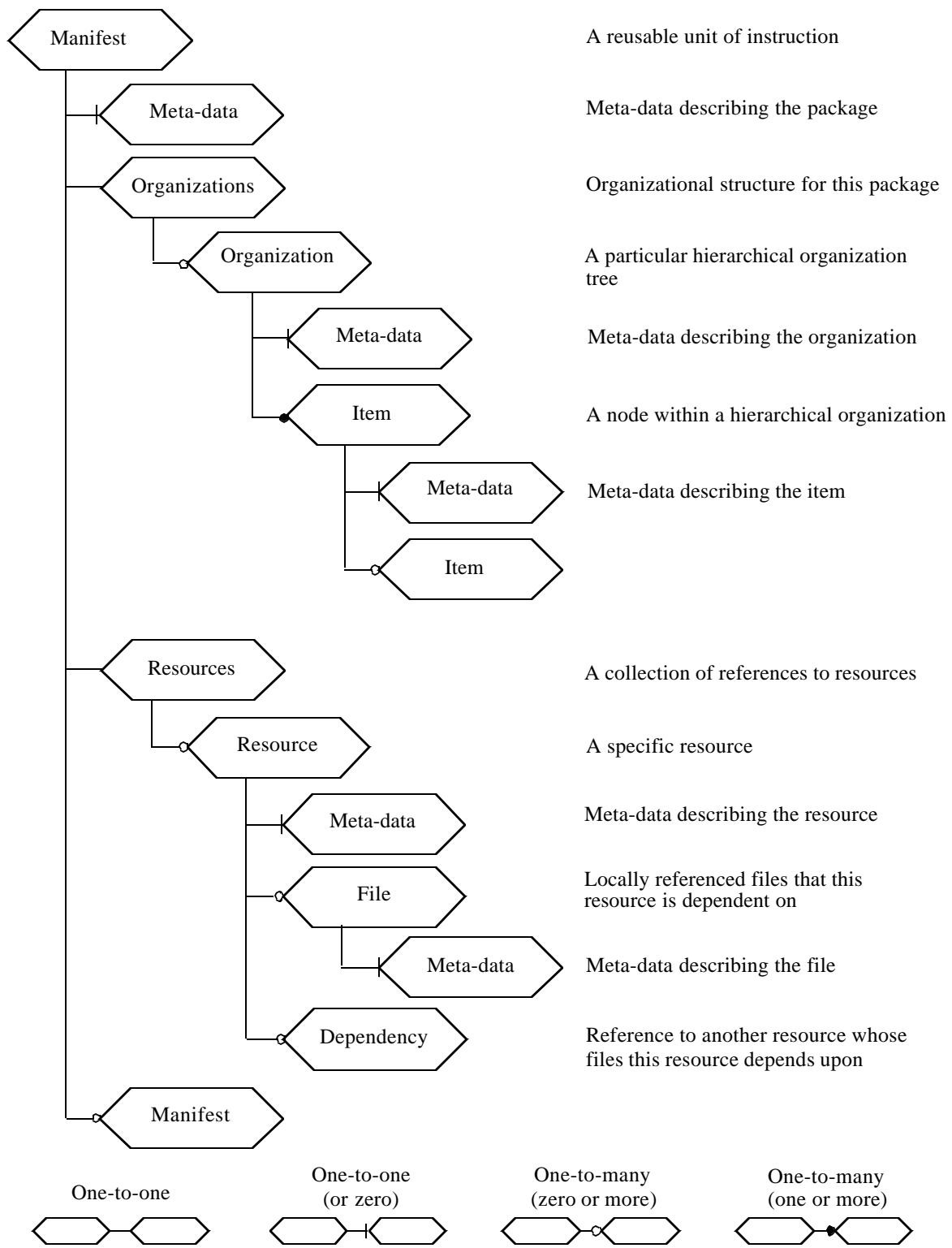


Figure 4.1 Manifest elements.

The table below provides a conceptual, informative description of the data objects. The columns used in the table refer to:

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.
Reqd:	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. C = conditional element, existence is dependent on values of other elements. O = optional element.
Mult:	Multiplicity of the element. Repeatability of an Element implies that all sub-elements repeat with the Element. Blank (-) = single instance. Number = maximum number of times the element is repeatable. n = multiple occurrences allowed, no limit.
Type:	A description of formatting rules for the data element: Type includes the maximum length of the element. The international character set specified by ISO 10646 will be used for all fields. Container = 'tag' element, of fixed length. ID = element used to uniquely identify an object. IDRef = a reference to an ID. String (n) = descriptive element (smallest permitted maximum). Boolean = True False.

Note: Additional descriptive information about the element.

- 1) In the table below, the Manifest elements contained in the Content Packaging Information Model are described using mixed case to enhance readability. Implementers of this specification should refer to particular binding specifications. For example, some XML bindings follow the W3C convention of using lowercase for all elements.
- 2) Elements surrounded by braces ({}) indicate areas in the Information Model where elements from other information models or specifications are expected to be included.

No.	Name	Explanation	Reqd	Mult	Type	Note
0	Manifest	A reusable unit of instruction. Encapsulates meta-data, organizations, and resource references.	M	-	Container	
0.1	Identifier	An identifier that is unique within the Manifest.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.
0.2	Version	Identifies the version of this Manifest (e.g., 1.0).	O	-	String (20)	Used to identify if there have been any changes to the Package. Identifier is the same in two Manifest files.
1	Meta-data	Meta-data describing the Manifest.	O	-	Container	
1.1	Schema	Describes the schema that defines and controls the Manifest.	O	-	String (100)	If no schema element is present, it is assumed to be "IMS Content".
1.2	SchemaVersion	Describes the version of the above schema (e.g., 1.0, 1.1).	O	-	String (20)	If no version is present, it is assumed to be "1.1"
1.3	{IMS Meta-Data }	This is where IMS Meta-data is inserted.	O	n	-	The information contained in this section is defined by the IMS Meta-Data specification.
2	Organizations	Describes one or more structures or organizations for this package.	M	-	Container	
2.1	Default	Indicates which organization scheme is the default one.	O	-	IDRef	If not supplied, the first organization element encountered is assumed to be the default.
2.2	Organization	Describes a particular hierarchical organization.	O	n	Container	Different views or organizational paths through the content can be described using multiple instances of organization.
2.2.1	Identifier	An identifier that is unique within the Manifest file.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.
2.2.2	Structure	Has a default value of "hierarchical" for describing the shape of an organization.	O	-	String (200)	Other values for structure will likely become part of a future specification.
2.2.3	Title	Describes the title of the organization.	O	-	String (200)	Used to help user decide which organization to choose.

No.	Name	Explanation	Reqd	Mult	Type	Note
2.2.4	Item	A node that describes the shape of the organization.	M	n	Container	Can be used in a hierarchical organizational scheme by ordering and nesting.
2.2.4.1	Identifier	An identifier that is unique within the Manifest file.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.
2.2.4.2	IdentifierRef	A reference to an identifier in the resources section or a (sub)Manifest.	O	-	String (2000)	
2.2.4.3	Title	Title of the item.	O	-	String (200)	
2.2.4.4	IsVisible	Indicates whether or not this item is displayed when the Package is displayed or rendered.	O	-	Boolean	If not present, value is assumed to be "true".
2.2.4.5	Parameters	Static parameters to be passed to the resource at launch time.	O	-	String (1000)	
2.2.4.6	Item	A sub-node within this organization.	O	n	Container	This is a sub-item and repeats all the parts of <item>.
2.2.4.7	Meta-data	Meta-data describing this item.	O	-	Container	See item 1. above
2.2.4.7.1	{IMS Meta-Data}	This is where IMS Meta-data is inserted.	O	n	-	The information contained in this section is defined by the IMS Meta-Data specification.
2.2.5	Meta-data	Meta-data describing this organization.	O	-	Container	See item 1. above
2.2.5.1	{IMS Meta-Data}	This is where IMS Meta-data is inserted.	O	n	-	The information contained in this section is defined by the IMS Meta-Data specification.
3	Resources	A collection of references to resources. There is no assumption of order or hierarchy.	M	-	Container	
3.1	Resource	A reference to a resource.	O	n	Container	
3.1.1	Identifier	An identifier that is unique within the scope of its containing manifest file.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.

No.	Name	Explanation	Reqd	Mult	Type	Note
3.1.2	Type	Indicates the type of resource.	M	-	String (1000)	The only current type is “webcontent”, defined as content that can be hosted in or launched by an Internet browser. This includes: - HTML-based content. - Content that requires plug-ins (e.g. Flash, Real Media.) - Executables that are launched by a browser.
3.1.3	HRef	A reference to a URL.	O	-	String (2000)	
3.1.4	Meta-data	Meta-data describing this resource.	O	-	Container	See item 1. above
3.1.4.1	{IMS Meta-Data}	This is where IMS Meta-data is inserted.	O	n	-	The information contained in this section is defined by the IMS Meta-Data specification.
3.1.5	File	A listing of files that this resource is dependent on.	O	-	Container	An element identifying a single file this resource is dependent on. Repeat as needed for each file for a given resource.
3.1.5.1	HRef	Identifies the location of the file.	M	n	String (2000)	
3.1.5.2	Meta-data	Meta-data describing this file.	O	-	Container	See item 1. above
3.1.5.2.1	{IMS Meta-Data}	This is where IMS Meta-data is inserted.	O	n	-	The information contained in this section is defined by the IMS Meta-Data specification.
3.1.6	Dependency	Identifies a resource whose files this resource depends upon.	O	n	IDref	This element identifies a single resource which can act as a container for multiple files that this resource depends upon.
3.1.6.1	IdentifierRef	A reference to an identifier in the resources section.	M	-	String (2000)	
4	Manifest	A reusable unit of instruction. Encapsulates meta-data, organizations, and resource references.	O	n	Container	See section 4.1 (Sub)Manifests below for more information.

4.1 (Sub)Manifests

When the identifierref of an <item> in an <organization> references a (sub)Manifest rather than another type of resource, this shall be interpreted as follows:

- If the (sub)Manifest does not include any <organization>, the reference cannot be resolved. This shall be treated as a null identifierref.
- If the (sub)Manifest includes an <organization> that can be used by the context in which the content is used, the root node of that organization (i.e., the <organization> element itself) shall merge with the <item> that references the (sub)Manifest, if the <item> becomes instantiated. If the same attribute is specified for both the <item> and the <organization>, but with different values, the value defined for the <item> manifest shall override the value defined for the <organization>.

The diagram in Figure 5.1 explains how the content of a (sub)Manifest is virtually merged with the content of a referencing manifest. The circles represent <items> in an <organization> structure.

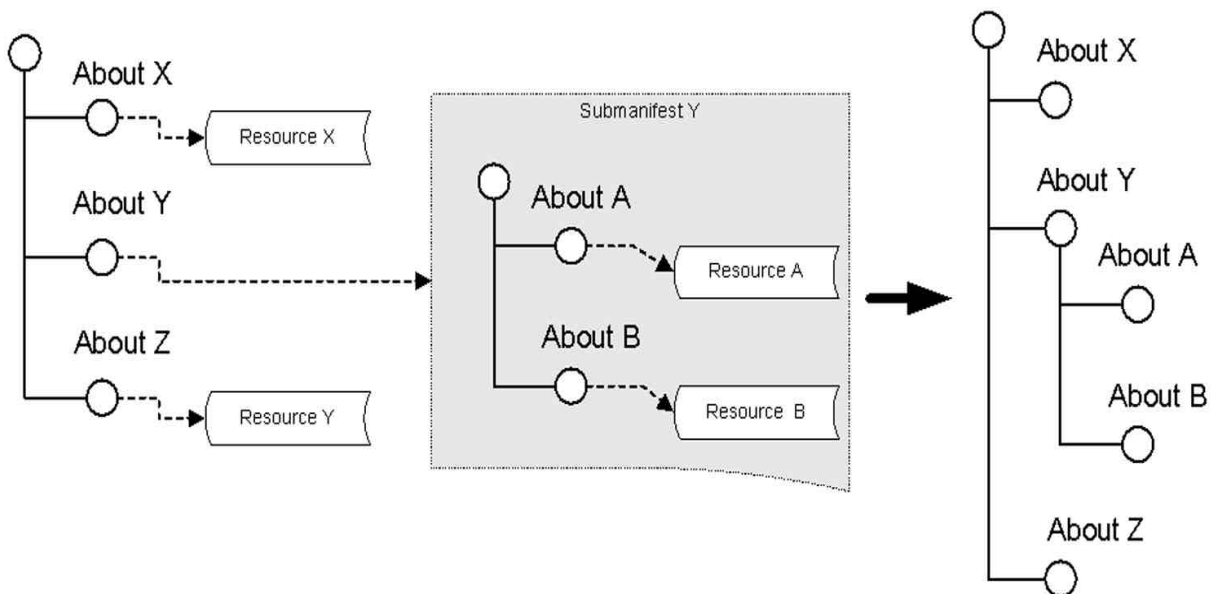


Figure 5.1 Merging <organization> from a (sub)Manifest.

In Figure 5.2, the diagram explains how the content of a (sub)Manifest is merged with an <item> that has children of its own, and how the referencing <item>'s children take precedence over those they merge with.

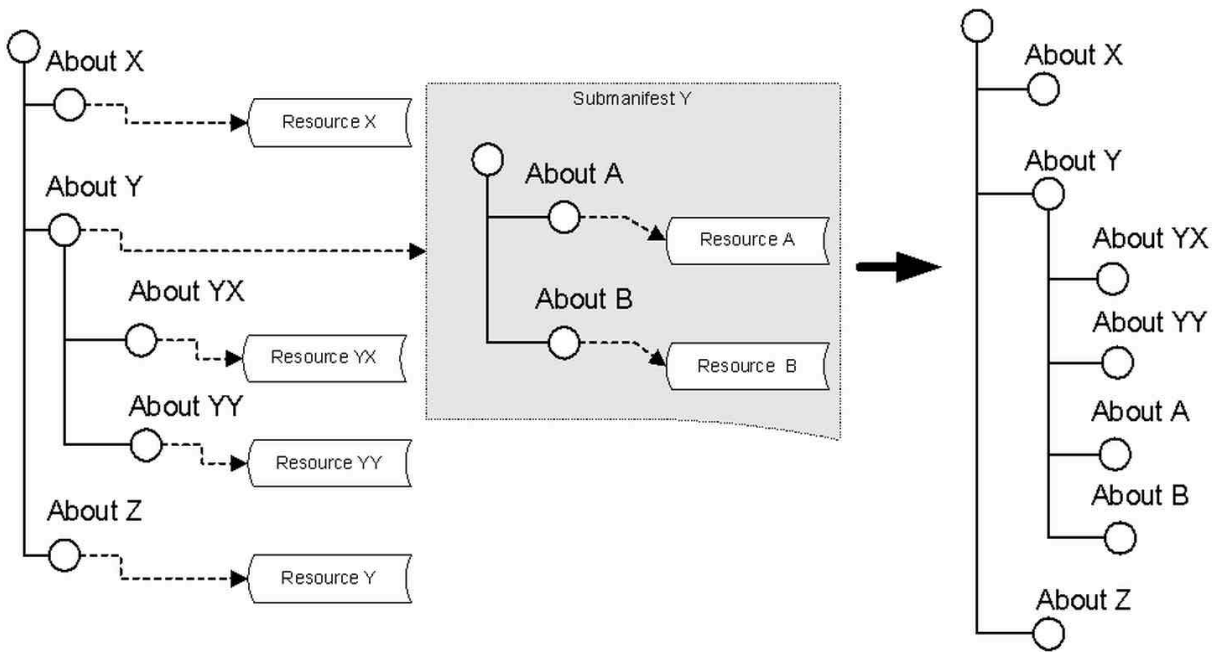


Figure 5.2 Merging <organization> from a (sub)Manifest when the referencing <item> has children.

Appendix A - List of Contributors

The following individuals contributed to the development of this specification:

Jay Beavers	Microsoft	Claude Ostyn	Click2Learn, Inc.
Adam Cooper	Fretwell-Downing	Mike Pettit	Blackboard
Rich Cushman	SCT	Daniel Rehak	Carnegie Mellon University
Philip Dodds	ADL	Tyde Richards	IBM
Steve Griffin	Eduprise	Udo Schuermann	Blackboard
Mike Halm	Penn State	Colin Smythe	IMS
Alan Hoberney	ADL	Schawn Thropp	ADL
Chris Moffatt	Microsoft	Tom Wason	IMS
Boyd Nielsen	NETg	Bill Young	Sun Microsystems
Bill Olivier	CETIS	Kenny Young	Microsoft

About This Document

Title	IMS Content Packaging Information Model
Editors	Thor Anderson, Mark McKell
Team Co-Lead	Adam Cooper (Fretwell-Downing)
Version	1.1.2
Version Date	August 2001
Status	Final Specification
Summary	This document describes the IMS Content Packaging Information Model, which is used to support content interoperability between different authors, publishers and other corresponding content developers.
Revision Information	8 August 2001
Purpose	Defines the Content Packaging Information Model
Document Location	http://www.imsglobal.org/content/packaging/

Revision History

Version No.	Release Date	Comments
Base 1.0	23 December 1999	The first formally released version of the full IMS Content Packaging Information Model Base Document.
Draft 0.9	8 February 2000	Draft of final version 1 specification accepted by IMS Technical Board
Public Draft 0.91	15 February 2000	Updated to address a) more consistent W3C-like handling of external files cleaner way to handle extended resource in the <resources> section
0.92	20 March 2000	Format updated with following changes: a) Move "invisible" attribute from <resource> element to <item> element b) Add <title> to <tableofcontents> c) Revert back to the <resource type="webcontent"> approach introduced in the v0.9 document d) Rename <organization> to <organizations>
1.0	2 May 2000	Final specification release presented to the IMS Technical Board: a) Adopted IMS meta-data version 1.1 b) Updated version information to 1.0
1.0	25 May 2000	Updated document to address the following open issues: a) Added a definition to the table for element 3.2.2 with the name "webcontent". b) Note #1 added about using mixed case to enhance table readability. c) Added section 4.8.1 on identifiers d) Note #2 added as description about how element names within curly braces are really just section placeholders and not real elements.
Public Draft 1.1	8 December 2000	Made minor text changes and updated the document to address the following issues: a) Replaced <tableofcontents> element with the <organization> element. b) Made <title> a sub-element of <item> rather than an attribute of it. c) Changed resource <item> element attribute "identifieref" to "resourceref". d) Made sub-level <manifest> a sub-element of <manifestref>, rather than <manifest>. e) Changed references from URL Base to XML Base. f) Reworded parts of section 2.1 in clarifying the definition of <organizations> and package. g) Added <dependency> element as a sub-element of <resource>. h) Updated XML samples.

Version No.	Release Date	Comments
Final 1.1	19 April 2001	<p>Updated document to address the following open issues:</p> <ul style="list-style-type: none"> a) Clarified the use of the <organization> and <item> elements. b) Added statement of recommendation to use PKZip v2.04g as the default Package Interchange File format in Section 1.2. c) Extended meta-data functionality to <organization>, <item>, and <file>. d) Changed the type attribute on <organization> to structure with a default value of hierarchical. e) Changed the href attribute on the <resource> element from Mandatory to Optional. f) Deprecated the use of <manifestref> and moved (sub)Manifests out of the <resources> block. g) Changed resource <item> element attribute back to “identifieref” from “resourceref”. h) Made several minor edits; changed references to sub-manifest to (sub)Manifest; updated the graphics. i) Changed Boolean type in the elements table from (0,1) to (true false), in order to work with schemas.
Update 1.1.1	23 May 2001	a) Updated XML-Schema sample in Binding Appendix B.
Final 1.1.2	8 August 2001	<p>Made several editorial corrections, and:</p> <ul style="list-style-type: none"> a) Updated references to IMS Meta-Data v1.2. b) Added explanation and diagrams to clarify the use of (sub)Manifests. c) Corrected the Manifest elements drawing.

Index

A

Aggregation 5

C

Co-Leads 15

Contributors 14

D

DTD 3

E

Editors 15

Elements

Default 9

Dependency 11

File 11

HRef 11

Identifier 9, 10

IdentifierRef 10, 11

IsVisible 10

Item 10

Manifest 9, 11

Meta-data 9, 11

Organization 9

Parameters 10

Resource 10

Resources 10

Schema 9

SchemaVersion 9

Structure 9

Title 9, 10

Type 11

Version 9

extensibility 6

I

IMS Manifest File 5

IMS Manifest file 4

imsmanifest.xml 4, 5

L

LMS 3, 4

M

Manifest 5, 7

O

Organizations 9

P

Package 4, 5

Package Interchange File 4, 5

R

Resource 5

Revision History 16

run time environment 3

S

scope 4

sub-Manifest 4, 5, 12

W

W3C 3

X

XML 3

Z

zip 4

IMS Global Learning Consortium, Inc. (“IMS”) is publishing the information contained in this IMS Content Packaging Information Model (“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 Content Packaging Information Model Revision: 8 August 2001