



IMS Content Packaging Information Model

**Final Specification
Version 1.0**

About This Document

Title	IMS Content Packaging Information Model
Editor	Thor Anderson
Version	1.0
Version Date	June 2000
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	June 2, 2000
Purpose	Defines the Content Packaging Information Model
Document Location	http://www.imsproject.org/content/packaging/cpinfo10.html

List of Contributors

The following individuals contributed to the development of this document:

Thor Anderson	IMS	Bill Olivier	UK IMS Centre
Jay Beavers	Microsoft	Mike Pettit	Blackboard
Philip Dodds	ADL	Tyde Richards	IBM
Steve Griffin	Eduprise	Udo Schuermann	Blackboard
Mike Halm	Penn State	Colin Smythe	IMS
Rich Cushman	SCT	Tom Wason	IMS
Chris Moffatt	Microsoft	Bill Young	Sun Microsystems
Boyd Nielsen	NETg		

Revision History

Version No.	Release Date	Comments
Base 1.0	December 23 rd , 1999	The first formally released version of the full IMS Content Packaging Information Model Base Document.
Draft 0.9	February 8 th , 2000	Draft of final version 1 specification accepted by IMS Technical Board
Public Draft 0.91	February 15 th , 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	March 20 th , 2000	Format updated with following changes: a) Move "isvisible" 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	May 2 nd , 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	May 25 th , 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.

Table of Contents

ABOUT THIS DOCUMENT	2
LIST OF CONTRIBUTORS	2
REVISION HISTORY	3
TABLE OF CONTENTS	4
1. INTRODUCTION	5
1.1 OVERVIEW	5
1.2 SCOPE & CONTEXT	5
1.3 STRUCTURE OF THIS DOCUMENT.....	5
1.4 REFERENCES.....	5
2. IMS CONTENT PACKAGING CONCEPTUAL MODEL	6
2.1 KEY ELEMENTS.....	6
2.2 STANDARD NAME FOR THE MANIFEST FILE.....	7
3. EXTENSIBILITY	8
4. MANIFEST ELEMENTS	9

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, 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, learning management systems and run time environments that have been developed independently by various software developers.

Note: The version 1.0 scope of the IMS Content Packaging specification is focused on defining interoperability between systems that wish to import , export, aggregate, and dis -aggregate packages of content. Future documents comprising the IMS Content specification will address requirements regarding content data models and communication between run time environments and learning management systems.

1.2 Scope & Context

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

- IMS Content Packaging XML Binding Specification v1.0;
- IMS Content Packaging Best Practice & Implementation Guide v1.0.

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. MANIFEST ELEMENTS: | The detailed description of the Manifest elements in terms of their properties and attributes; |

1.4 References

- [Content, 00a] *IMS Content Packaging XML Binding Specification*, T. Anderson, Version 1.0, [IMS](#), May 2000.
- [Content, 00b] *IMS Content Packaging Best Practice & Implementation Guide Document*, T. Anderson, Version 1.0, [IMS](#), May 2000.

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 [Content, 2000b], this is part of the larger IMS Content Framework, which forms the basis for this and future specifications.

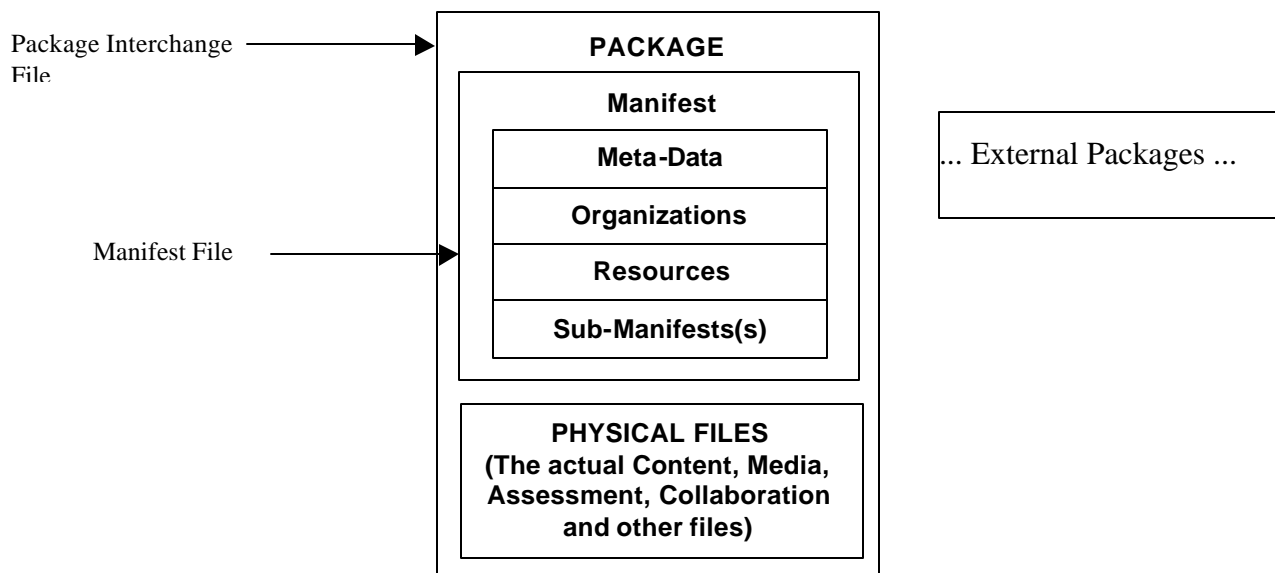


Figure 2.1 Package representing version 1.0 scope.

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.

Package – a logical directory, which includes a specially named XML file, any XML control documents it references (such as a DTD, XDR, or XSD file) and subdirectories 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 one or more 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-Manifests** - one or more optional, logically nested Manifests.

- **Physical Files** these are the actual media elements, text files, graphics and other resources in their various subdirectories 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, an entire course, or a collection of courses. A package must allow itself to be aggregated or disaggregated into other packages. A package must be able to stand alone; that is it contains 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, XDR, 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 also contains one 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. It is left up to content developers to describe their content in the way they want it to be considered for aggregation or dis-aggregation. 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 at which the sub-manifest is scoped: course or instructional ‘object’.

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 producer 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 packager that 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 may contain.

Resource – Physical files are the resources that consist of media files, text files, assessment objects or other pieces of data in file form. The combination of Resources is generally categorized as "content". Unlike a Package, a resource is not stand-alone. Each physical Resource may be described in a Resource element within a manifest’s XML or listed as a file supporting a Resource. The actual Resources described in the XML file are included with it as part of the set of files comprising a package.

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** :

```
imsmanifest.xml
```

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.

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

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 1.0 Meta-data specification, it defines only the basic structures for organization (TableOfContents) 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.

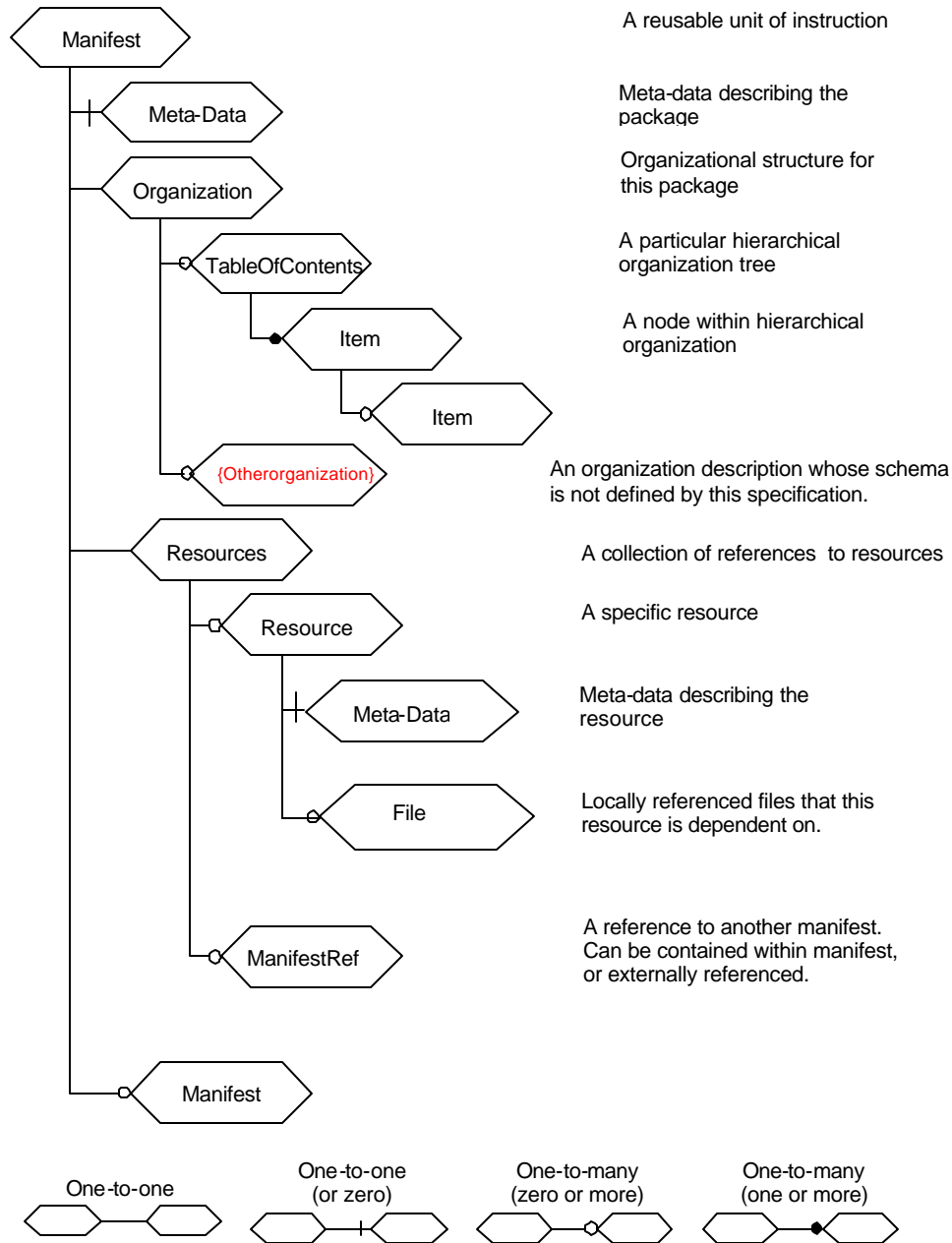


Figure 4.1 Manifest elements.

Table 4.1 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. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • Container = 'Tag' element, of fixed length • ID = element used to uniquely identify an object. • IDRef = a reference to an ID. • String (n) = descriptive element • Boolean = binary flag (O/1)
Notes:	Additional descriptive information about the element.

Notes:

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.

Table 4.1 Manifest Elements.

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 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 version of the above Schema, e.g. 1.0, 1.1.	O	-	String (20)	If no version is present, it is assumed to be "1.0"
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 version 1.1.
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 TableOfContents element encountered is assumed to be the default.
2.2	TableOfContents	A particular hierarchical organization.	O	n	Container	Different views or organizational

No	Name	Explanation	Reqd	Mult	Type	Note
						paths through the content can be described using multiple TableOfContents
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	Title	Title of the TableOfContents.	O	-	String (256)	Used to help user choose which organization to choose.
2.2.3	Item	A node within this organization.	M	n	Container	
2.2.3.1	Identifier	An identifier that is unique within the manifest file.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.
2.2.3.2	IdentifierRef	A reference to an Identifier in the Resources section.	O	-	ID	
2.2.3.3	Title	Title of the Item.	O	-	String (256)	
2.2.3.4	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 "1".
2.2.3.5	Parameters	Static parameters to be passed to the resource at launch time.	O	-	String (1024)	
2.2.3.6	Item	A sub-node within this organization.	O	n	Container	This is a sub-item and repeats all the parts of Item.
2.3	{Other Organization}	An organization description whose type and schema is not defined by the specification is inserted here.	O	n		
3	Resources	A collection of references to resources. There is no assumption of order or hierarchy.	M	-	Container	

No	Name	Explanation	Reqd	Mult	Type	Note
3.1	URL Base	This provides a relative path offset for relative URLs in the Package.	O	-	String (2048)	
3.2	Resource	A reference to a resource.	O	n	Container	
3.2.1	Identifier	An identifier that is unique within the manifest file.	M	-	ID	See the Best Practice Guide for guidelines on the use of identifiers.
3.2.2	Type	Indicates the type of resource.	M	-	String (20)	<p>The base type is “webcontent”.</p> <p>“webcontent” is defined as content that can be hosted in, or launched by an internet browser.</p> <p>This includes:</p> <ul style="list-style-type: none"> - HTML-based content. - Content that requires plug-ins (e.g. Flash, Real Media.) - Executables that are launched by a browser.
3.2.3	URL Base	This provides a relative path offset for the resource file(s).	O	-	String (2048)	
3.2.4	HRef	A reference to a URL	M	-	String (2048)	
3.2.5	Meta-data	Meta-data describing this resource	O	-	Container	See item 1. above
3.2.5.1	{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 version 1.1.
3.2.6	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

No	Name	Explanation	Reqd	Mult	Type	Note
						for each file for a given resource.
3.2.6.1	HRef	Identifies the location of the file.	M	n	String (2048)	
3.3	ManifestRef	A reference to other manifest elements that the referring Manifest depends upon. It can be contained within the Manifest file, or externally referenced.	O	n	Container	
3.3.1	IdentifierRef	Resource identifier that is used to resolve the ultimate location of a resource within the same manifest element.	M	-	ID	
4	Manifest	Another reusable unit of instruction contained within the top-level manifest element.	O	n	Container	

Index

A		
ADL.....	2	
Attributes		
Title	2	
C		
Content packaging 1, 2, 3, 5, 6, 7		
D		
Default	11	
E		
Elements		
Default	11	
File	6, 7, 13	
Identifier.....	11, 12, 13	
IdentifierRef	12, 14	
IsVisible	12	
Item	12	
Manifest.....	5, 6, 7, 9, 11, 14	
ManifestRef.....	14	
Meta-data.....	6, 11, 13	
Organizations	6, 11	
Parameters	12	
Resource.....	7, 13, 14	
Resources	6, 12	
Schema	11	
SchemaVersion	11	
TableOfContents.....	11, 12	
Title	2, 12	
URL Base.....	13	
Version	1, 2, 3, 5, 11	
F		
File	6, 7, 13	
I		
Identifier	11, 12, 13	
IdentifierRef.....	12, 14	
IMS	1, 2, 3, 5, 6, 7, 11, 13	
Information model.....	1, 2, 3, 5, 6	
Information Model....	1, 2, 3, 5, 6	
ISO 10646.....	10	
IsVisible	12	
Item.....	12	
M		
Manifest.....	5, 6, 7, 9, 11, 14	
Manifest sub-elements		
Meta-data.....	6	
Resources	6, 12	
ManifestRef.....	14	
Meta-data		
Status.....	2	
Title	2	
Version.....	1, 2, 3	
Meta-data.....	3, 6, 11, 13	
O		
Organizations	6, 11	
P		
Package	6, 7, 11, 12, 13	
5, 6, 7		
Parameters	12	
R		
Resource.....	7, 13, 14	
Resources	6, 12	
S		
Schema	11	
SchemaVersion	11	
Specifications		
ADL.....	2	
IMS.....	1, 2, 3, 5, 6, 7, 11, 13	
Information Model ..	1, 2, 3, 5, 6	
XML Binding.....	5	
Standards		
ISO 10646.....	10	
W3C.....	3	
T		
TableOfContents.....	11, 12	
Title	2, 12	
U		
URL	13	
URL Base.....	13	
V		
Version	1, 2, 3, 5, 11	
X		
XML		
ELEMENT	10	
XML Binding.....	5	