Best Practice and Implementation Guide

William Reilly, Robert Wolfe
{wreilly, rwolfe} @ mit.edu

MIT Libraries: Digital Library Research Group & Metadata Services Unit

Version 1.0, January 19, 2005

http://cwspace.mit.edu/docs/WorkActivity/Metadata/Packaging/IMS-CP/profile/profile10/profileDoc/cwspace-imscp-docv1p0.pdf
INTRODUCTION ........................................................................................................... 4
Relation to IMS Content Packaging documentation .............................................. 4
This Document: Versions 0.1, 1.0, 2.0 .................................................................... 4

CONTENT PACKAGE .................................................................................................. 5
Standard name for the Manifest File ....................................................................... 6
IMS Standards .......................................................................................................... 6

OBJECT MODEL ......................................................................................................... 7

PROFILE DEVELOPMENT ISSUES, NOTES .......................................................... 8

IMS Extensions ......................................................................................................... 8
Version 1.0 Statement (vs. 2.0) ................................................................................ 8
Package Type: Single Course ................................................................................... 8
<profile> .................................................................................................................... 9
<profileVersion> ....................................................................................................... 9
<checksum> ............................................................................................................. 9
<sourceSystem> ...................................................................................................... 9
<manifest @packageType> ...................................................................................... 9
<ocw:sectionTemplateType> .................................................................................. 9

Object Model ............................................................................................................ 10
Nested <item>s ......................................................................................................... 10
Multiple <organization> .......................................................................................... 10
Sub-Manifests .......................................................................................................... 11

Globally Unique Persistent Identification ............................................................. 11
Item and Resource IDs, IDREFs, HREFs ................................................................. 12
OCW Identifier Construction Details .................................................................... 13

Standard Schemas ................................................................................................... 16
IMS-CP ..................................................................................................................... 16
IEEE LOM ............................................................................................................... 16
ADL SCORM CP ...................................................................................................... 16

Extension Schemas .................................................................................................. 17
cwspace_imscp.xsd ............................................................................................... 17
ocw_imscp.xsd ........................................................................................................ 17

Miscellaneous Topics .............................................................................................. 17
<iitem><title> Values ............................................................................................... 17
OCW Global Artifacts ............................................................................................ 17
PROFILE INFORMATION MODEL AND BEST PRACTICES WITH XML BINDING SPECIFICATION

<manifest> Element ........................................................................................................ 18
  XML Example........................................................................................................ 18
  Best Practices ......................................................................................................... 19

<metadata> Element ..................................................................................................... 20
  Best Practices .......................................................................................................... 20

<metadata> Element Extensions .................................................................................... 20
  XML Example.......................................................................................................... 21
  Best Practices .......................................................................................................... 22

<organizations> Element .............................................................................................. 23
  XML Example.......................................................................................................... 23
  Best Practices .......................................................................................................... 23

<organization> Element ............................................................................................... 24
  XML Example.......................................................................................................... 24
  Best Practices .......................................................................................................... 24

<item> element ............................................................................................................. 24
  XML Example.......................................................................................................... 25
  Best Practices .......................................................................................................... 26

<resources> Element .................................................................................................... 26
  XML Example.......................................................................................................... 26
  Best Practices .......................................................................................................... 26

<resource> Element ..................................................................................................... 26
  XML Example.......................................................................................................... 27
  Best Practices .......................................................................................................... 28

<(sub)manifest> Element .............................................................................................. 28
  XML Example.......................................................................................................... 28
  Best Practices .......................................................................................................... 29

Sample XML Files ........................................................................................................... 30
Introduction

Note: This document (Version 1.0, January 19, 2005) supersedes the incomplete, January 7, 2005, version 0.1, which was effectively an outline of this profile documentation. This document accompanies the other 1.0 CWSpace IMS-CP Profile deliverables which were posted on January 7, 2005.

CWSpace (http://cwspace.mit.edu), an MIT iCampus sponsored project, has as its principal goal the successful archiving of courseware materials from MIT’s OpenCourseWare (OCW) into MIT’s digital institutional repository system DSpace. The project’s charter is also to ensure the subsequent interoperability of this teaching and learning material among learning management systems at MIT (e.g. Stellar, SloanSpace, and (eventually) Sakai, etc.). To that end, the selection of the Content Packaging standard from IMS—the de facto standards organization for technologies in higher education—was seen as the best choice for a data interchange format, to move the content between these various learning, publishing, and repository systems.

This Profile documentation, which contains details on how to use the IMS Content Package (IMS-CP) specification, is intended for the staff of systems that seek to deposit and retrieve courseware content from MIT’s DSpace, which, as an outcome of the CWSpace project, will be developed into a “learning object repository” (LOR), according to this specification. As noted, systems that use this Profile would additionally benefit from being able to exchange courseware content with one another directly (outside DSpace).

While the Profile has in large part been driven by the needs of OCW, efforts have been made to include the use cases and requirements of the MIT LMSs. In fact it is hoped that the resulting specification is sufficiently general to permit wider adoption, and that it is sufficiently extensible to allow further profile development.

Relation to IMS Content Packaging documentation

This profile of IMS’s Content Package specification does not seek to recreate the IMS documentation, which includes a Best Practices and Implementation Guide, an Information Model, and an XML Binding specification. Instead, this document provides descriptive information on those parts of the IMS-CP for which the CWSpace intended use has either a constraint, usage note, or extension. Note that this single document contains both a “Best Practices and Implementation Guide” and the Information Model (there is no XML Binding document per se for CWSpace, though important annotated XML fragments are included within the Information Model). There are also an external schema for CWSpace, a second external schema for OCW, sample XML (imsmanifest.xml), an object model graphic, and a list of the files in the sample package, to complete the list of artifacts in the overall profile.

It should also be noted that a related effort was conducted by OCW and Stellar in early 2004, and a specification was prepared for what was named the “Content Data Interchange Format” (CDIF). This document was derived in part from that specification.

This Document: Versions 0.1, 1.0, 2.0

Version 0.1 = initial delivery of this document, January 7, 2005. Outline completed, critical content describing Item and Resource @identifier and @identiferrefs completed.

Version 1.0 = follow-on delivery January 19, 2005, to flesh out other narrative content. This version does not alter other 1.0 deliverables of January 7, 2005 (Sample XML manifest file, external schema,
list of files in package). Version 1.0 can be said to focus on the practical short- to mid-term present capabilities of OCW systems, practices, and policies.

Version 2.0 = placeholder name ("work-in-progress") for next version of Profile (intended for post CWSpace Year One = after April 2005). Version 2.0 documentation can be said to the more comprehensive expression of how to use this profile, with more generalized guidance for systems beyond MIT OCW.

**Content Package**

[Note: This section largely taken directly from CDIF, which in turn based much from the IMS-CP documentation.]

A ‘Content Package’ represents a unit of content. It might be content for an entire subject, or part of a subject. The content package contains all files (content and control) and the associated directory structure. The content files are the actual physical resources, and the control files are XML control documents. A content package must exist as a stand-alone entity; that is, it must contain all the information needed to import the subject when it has been unpacked.

The content package, as depicted in the following figure consists of three major elements:

- Package Interchange File
- Manifest File
- Physical content files

**Package Interchange File** – a single file, (e.g., .zip, .jar, .cab) which includes the manifest file and all other physical files as identified by the manifest. The 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. Once a package has been incorporated into a single file for transportation, it is called 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. The manifest file and any other supporting XML files required by it (DTD, XSD) must be at the root of the distribution medium.

**Manifest File** – The manifest file is an XML file that describes the subject content and organization in the context of a ‘manifest’. It is a description of the resources in the package. Without this file, the package cannot be processed. 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 is a mandatory XML
element describing the subject. The nested sub-manifests describe the sections in the subject such as home page, syllabus, calendar, problem sets, etc. The manifest is made up of the following sections:

- Metadata section – an XML element describing a subject 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 metadata describing the resources, and references to any external files.
- Sub-manifest – one or more nested manifests (sections).

A manifest may contain multiple descriptions of how the content may be organized for presentation, which would allow the previewing of the content package via a toolkit such as the LRN toolkit.

The 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). The XML can also refer to other schema definition files for namespace and definitions for the elements.

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). Resources may also include assets that are outside the package but available through a URL, or collections of resources described by manifests. 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.

**Standard name for the Manifest File**

Content distributed according to the specification must contain a manifest file. To ensure that the manifest file can always be found within a package, it has a pre-defined name and location (the root of the Package Interchange File (PIF)):

/imsmanifest.xml

The file name enables IMS-CP-aware software to automatically read the content of the content packages. It is required that the name be kept, as above, in all lowercase letters.

**IMS Standards**

- The specifications outlined in this document follow the IMS specifications. The CWSpace IMS-CP profile extends the IMS specs in such a way that the IMS-CP-aware systems will be able to read Content Packages that are 100% IMS compliant also. The extensions to IMS are limited to the few elements added via the cwspace_imscp.xsd file as well as the ocw_imscp.xsd file (see below).
Object Model

The object model shows the high level objects in the package.

- **Manifest**: The highest-level manifest node represents a subject. There will be only one root level manifest node in the content package.

- **Metadata**: The metadata for the root level manifest node will contain the subject level metadata. All the IP related information is captured with the metadata.

- **Organizations**: This describes the organizational structure of the subject. This is a container that contains the sections and in what order they appear.
• **Organization**: This shows a particular hierarchical organization tree.
  
  • **Item**: Each Item refers to a section in the subject. The order in which
  the items appear within the organization, determines the ordering of
  the sections within the subject.

• **Resources**: This is a container that represents a collection of the references to resources.
  
  • **Resource**: This indicates a specific resource. The resource can be any piece
  of content within the file.

  • **Metadata**: This is the metadata associated with a resource. All the IP
  related information is captured with the metadata.

### Profile Development Issues, Notes

**IMS Extensions**

**Version 1.0 Statement (vs. 2.0)**

It is important to note that for much of the material covered in this Version 1.0 of the profile document, the immediate practical information primarily concerns the deposit and retrieval of OCW courses into and out of DSpace. Also under consideration here is the extension of that basic use case to other MIT LMSs: at a minimum the obtaining of an OCW course from DSpace, with the next level of profile adoption to be the packaging up of Stellar or SloanSpace courses for deposit into DSpace. This last may be beyond Version 1.0, and would be covered in greater depth in Version 2.0 (post CWSpace “Year One” (April 30, 2005).

**Package Type: Single Course**

*Note: candidate “types” for this discussion include: single course, multiple courses, single “learning object” (LO), multiple LOs, hybrid (course(s) and LO(s)).*

As an IMS-CP imsmanifest.xml file may describe any arbitrary collection of content, we must define what kind(s) of package collections are intended for use with this Profile, within the CWSpace sphere of courseware exchange.

For Version 1.0, the package type is simply one entire course. This “one course” model straightforwardly follows the hierarchical organization of the OCW public website. Learning objects are expressly not going to be represented separately nor even identified within the single manifest (though this approach had been considered).

The CWSpace project use cases to date have not identified the need to package multiple courses simultaneously, though future work should permit this. [Worth mention is concurrent Web Services development on behalf of CWSpace (by the University of Cambridge’s Centre for Applied Research in Educational Technologies (CARET)); they may provide a very simple wrapper mechanism for accommodating this.] The eventual use of proper IMS-CP sub-manifests for this purpose would be brought into play when this use case is more fully addressed. Note that most if not all of the challenges of using sub-manifests that were encountered when trying to provide for LOs along with their parent course (as discussed below) would not apply in the case of aggregating multiple courses using sub-manifests.

Single LOs from OCW for CWSpace Year One deliverables are not likely to be in a state of readiness for automatic processing to IMS Content Packages. The current intent instead is to prepare (non-automated) example LOs as stand-alone IMS-CPs (that is, not integrated with a Content Package for the course they are found in). There is also other related work underway on OCW systems and
workflows, to assist in the identification, file inventorying, metadata tagging, and capture of structure of LOs, such that at least some of these objects will be ready for automated processing to IMS Content Packages, according to the same principles as delineated in this IMS-CP Profile for entire courses.

The aggregation of multiple LOs into a single IMS-CP is a step that would naturally (and easily) follow the case of single LOs. Akin to the similar note above (re: sub-manifests for multiple courses), the use of sub-manifest for this task of multiple (sibling, parentless) LOs would be more straightforward than it was when we tried using sub-manifests for LOs in the same overall manifest as their parent course. Again, see discussion below re: why sub-manifests are not in use for this Version 1.0 of the Profile.

Finally, the "hybrid" manifest of some permutation of more than one course (perhaps) paired with one or more LO is a case which should become better defined as we work through the other options described above. The key distinction to make, perhaps, concerns whether the LOs present are by definition "children" of the course(s) in the manifest (as opposed to LOs from other courses entirely). The duplication of XML ID attributes (not permitted) in such a case is potentially problematic, and will require further study.

<profile>

This extension element serves as a declaration that “this XML document is not only of the type ‘IMS-CP,’ but is further in compliance with the Profile for IMS-CP as defined by CWSpace.”

<profileVersion>

This extension element permits the precise version number of that profile to be recorded. This usage is akin to the IMS-CP <schema> and <schemaVersion> elements.

<checksum>

This extension element is for the moment reserved for future refined definition of usage practice. It is currently optional, and not used.

<sourceSystem>

This extension element is intended to capture the name of the system (e.g. an enumeration containing OCW, Stellar, SloanSpace, DSpace and MetaMedia) that served as the source of the content.

The interpretation here is meant to reflect the most recent “value-add” system (that is, a course originally obtained from Stellar that OCW then heavily processes to become an OCW publication course website would, upon export from OCW, get the label “OCW” in this sourceSystem element, not Stellar). More thinking will undoubtedly have to occur in this area, especially with the more fully addressed issues in the Lifecycle use cases for courses, PIFs, manifests, and similar.

<manifest @packageType>

This extension attribute serves as the enumerated selection of types; to date simply “Course” and the (not yet used in Version 1.0) “LO”. Others may be added (via schema edits).

<ocw:sectionTemplateType>

This extension attribute belongs to a separate extension schema, one owned by OCW and reflecting the more OCW-specific needs for extensibility of the IMS Content Package and the IMS-CP Profile for CWSpace.

The information in this element reflects the named templates that OCW uses for page layout choice and similar in their page preparation engine.
Object Model

Nested <item>s

The generic IMS specification for the <item> element in the Content Package is simply that these can nest. It is therefore a completely generalized container, useful for accommodating all kinds of content and content relations, but perhaps a bit too loose to help the designer of information packages to know how best to apply them to a particular use.

In the case of the CWSpace IMS-CP Profile, the "particular use" of course is the capture of entire OCW courses. Therefore, for this Version 1.0 document, the somewhat admittedly OCW-centric application of the usage for these 'item' elements is driven by OCW's own object model and information architecture. In practice, this is most clearly seen in the values recorded in the "identifier" attribute of the item element. Highly detailed instructions on how to determine that precise value are given below. Here the intent is to simply provide a brief description of some of OCW's terminology and practice in organizing its content, as is then used in creating item elements (and nested item elements) in the IMS Content Package.

As can be seen in the graphic for the Object Model in this document, the OCW Course is sub-divided into what are termed "Sections"; these HTML pages can contain hyperlinks directly to OCW "Resources" (e.g. .PDF, .JPG, etc.), or to other HTML pages that at OCW are termed "Detail HTML Pages." These Detail Pages are similar in that they too can contain hyperlinks directly to OCW Resources, or, in nested fashion, to yet another Detail HTML Page. All of these artifacts described in this paragraph are classed as <item>s in the IMS Content Package. They are nested as items as follows the logical, hierarchical set of links found in the website. Their "identifier" attribute (an XML "ID") is constructed to also reflect that hierarchical relationship (as opposed to only following the file's path location). As stated above, the details on this are found below, but the point here is to outline the OCW approach and terminology to this organization.

The expectation is that this approach, while described and mapped to one system's terminology (e.g. OCW's) for website construction in Version 1.0, should survive and be straightforwardly generalizable in Version 2.0 to other system's approaches (e.g. Stellar, et al.).

Multiple <organization>s

This is to note that while the basic Version 1.0 Profile presents a single <organization> element to capture the essential logical presentation of the OCW website, the same Profile (even in Version 1.0) can support multiple <organization> elements to reflect other ways in which the same content might be arranged.

For example, an OCW course archived to DSpace might be subsequently obtained by the Stellar LMS, inside of which additional or different relationships amongst the content files might be established. Stellar uses the principle of "arcs" to apply metadata labels to files, allowing for different dynamic views (e.g. by kind, by date, by topic). These different views might be represented as additional <organization>(s) in a subsequent IMS Content Package then created by Stellar, for deposit to DSpace or possibly direct exchange with OCW.

As another example, the eventual adoption of including LOs in the same manifest as complete courses would present another <organization> to contain those LOs apart from the <organization> holding the course. Most likely, any collection of LOs would have the straightforward and flat relationship of being merely a set of siblings, though this is not prescribed. [Note also that LOs can nest (e.g. an HTML textbook LO containing a Java applet simulation LO inside one of its chapters).]

The nature of which <organization> is the default one is significant. The IMS-CP specification indicates this can be specified on the parent <organizations> element in the "default" attribute. Lacking that (it's optional), the first child <organization> encountered is the default. From the CWSpace view, even taking into account Stellar and SloanSpace possible requirements, to keep the
source OCW organization as the default would appear to make the most sense, and is recommended here.

**Sub-Manifests**

The IMS-CP Profile in Version 1.0, as it will be used in practice for the purpose of OCW courses, does not use the IMS Content Package sub-manifest concept. The Version 1.0 Profile per se does not expressly prohibit the use of sub-manifests, and it may be that content packaging activity at the LMSs or possibly even DSpace might employ sub-manifests for some yet to be determined reason. Version 2.0 of the Profile indeed will document the anticipated future kinds of uses of this mechanism.

Generally speaking, the use of sub-manifests is intended for content that is expected to be disaggregated automatically, usually upon receipt or import of the package. By definition, the contents of any sub-manifest must be entirely self-sufficient and be able to stand alone (e.g. no dependencies outside the sub-manifest). A related implication with the use of sub-manifests is that in their status as sibling (or co-existent, anyway) members of a larger, overall manifest, any use of XML ID attributes must be assured of being unique throughout the manifest. Therefore, any one resource that might for any reason be included in more than one sub-manifest would have to be duplicated in each, with a unique (different) ID in each. This holds potential for difficulties.

The current status at OCW is that LOs are not identified within the system, and therefore the automated authoring of sub-manifests for them is not feasible. As noted above, plans are in place to create new tools and processes at OCW for the identification, file inventorying, metadata tagging, and capture of structure of LOs, which would lead to the possibility of creating IMS-CPs for LOs.

Two additional points are worth mention as to why, even beyond the technical impediment noted above regarding the identification and automated processing of LOs, sub-manifests are not in use for LOs with OCW course Content Packages. First, the problem concerning one resource being used in more than one LO presents the difficulty described just above owing to the uniqueness required of XML ID attributes. The risk of confusion engaged in assigning two different ID values to what is really the same resource appears to be high enough to steer the team away from using sub-manifests for that reason alone. Secondly, the gains afforded content package maintenance by the re-factoring out of commonly referenced resource dependencies (e.g. the dozen or so files in the grouping termed, "webrenderrequirements" (e.g. .CSS, logo.GIF, .JS, etc.)) is lost when you employ sub-manifests, because again each sub-manifest needs to be self-sufficient, and so needs its own self-contained set of webrenderrequirements information, even though that may be a perfect duplicate of the same information in all the other sub-manifests you may have.

**Globally Unique Persistent Identification**

The root element manifest has an “identifier” attribute that the CWSpace Profile (for OCW) indicates should be populated with a Uniform Resource Name (URN). Note that for non-OCW systems, other unique identifiers may come into play, for example Handles, DOI, etc.

URN = Uniform Resource Name, is the scheme of choice for IMS. It is comprised of two parts in addition to the "URN" prefix: the Namespace Identifier or NID, which is intended to be a formally registered string staking out a domain for organizations using URNs, the Namespace Specific String or NSS is the domain specific and unique string that completes the ID.

Handles and Digital Object Identifiers or DOIs are other examples of globally unique, persistent IDs they function the exact same way the the URN, constructed of two parts: a first part, globally registered that defines an organizational domain and the second providing a domain specific and unique ID.
Item and Resource IDs, IDREFs, HREFs

Inside the manifest, the unique identifiers do not have to possess the same formal degree of name collision protection as the ID for the manifest overall (e.g. URN etc.), but the techniques below still employ very URL-like values, to ensure that the degree of uniqueness of these resources and items is going to be sustainable more widely than simply inside the manifest of origin. These objects may well be disaggregated and re-assembled with other content, and it would be highly useful if the identifiers were assured of continuing to be unique, even in unanticipated contexts.

Therefore, as we look at the key identifying attributes inside the manifest, it’s important to understand how to properly use the established naming conventions for the @ID, @IDREF, and @HREF attributes on the <item> and <resource> elements as used in the IMS Content Package.

To take a “bottom-up” approach to gaining this understanding, the next paragraphs walk through a sequence showing how they are related.

Note that here in the Version 1.0 Profile document, much of this information is OCW-centric, nevertheless the same principles can be applied to the HREFs and naming conventions of other systems.

First, the value of the <file> element’s @href attribute is, for obvious reasons, driven directly by the actual path and filename value of the file as stored in the package. For this path and filename we have chosen to follow complete paths from the OCW website root (excluding however the leading “/”, such that the set of files remain linked in a relative way, rather than absolute). This approach is recommended for all systems (not just OCW).

Then, the value of the parent <resource> element’s @identifier attribute is based on the same string as the @href on the primary entrypoint <file> element. (It is processed to look a little different than a URL (with ”/” slash marks) by substituting instead periods “.” for each slash mark.) To essentially re-use the URL path as @identifier was felt to be the best choice for 1) assured unique identification of the resource, across manifests and within a particular domain, and 2) ease of adoption by the OCW development staff.

Moving up “the stack,” as it were, the value therefore of the <item> element’s @identifierref attribute is, by definition, the exact same value as the <resource> element’s @identifier. So once again we have essentially the URL path.

Finally, the value of the <item> element’s @identifier attribute differs from the otherwise URL path-centric approach used in the preceding three attributes. The <item> @identifier presents us with the opportunity to reflect something less driven by file system placement, and more indicative of (one of) the ways in which the content is organized in an information architecture at the presentation level. This means we can provide a human readable (and comprehensible) directory path that makes logical sense for content organization and linking.

As an example, resources (e.g. PDFs like 418hw4.pdf) organized “beneath” the Assignments section are in fact stored at /NR/Course/…GUID/418hw4.pdf. But our <item> @identifier need not reflect that more arcane and system maintenance related file location, and instead can depict the natural hierarchy as established to the user by means of hyperlinks, as in …Course.Assignments.418hw4.pdf. And further, the site’s organization may also permit access to many of the same PDFs from the Calendar section. The <item> @identifier from that page (leading to the same <resource>) would be …Course.Calendar.418hw4.pdf.
OCW Identifier Construction Details

The tables below delineate OCW-specific methods for deriving ID attribute values. Similar details would need to be determined and documented for additional systems wishing to prepare IMS-CP packages to meet the CWSpace IMS-CP Profile. The principles for doing so would follow the general guidelines established here.

### Manifest Identifier Scheme

**Object Model Node**: `/manifest/@identifier`

<table>
<thead>
<tr>
<th>Object Model Node</th>
<th>Identifier Scheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manifest (OCW Course)</td>
<td>&quot;urn&quot; + &quot;&quot; + &quot;edu.mit.ocw&quot; + &quot;&quot; + [Channel Name]</td>
<td>urn_edu.mit.ocw_21H-418Technologies-of-Word-1450-2000Fall2002</td>
</tr>
</tbody>
</table>

**Preferred URN:** urn:edu.mit.ocw:21H-418Technologies-of-Word-1450-2000Fall2002

**Compromise URN:** urn_edu.mit.ocw_21H-418Technologies-of-Word-1450-2000Fall2002

**Discussion:** URNs use the colon `:` but (unfortunately) XML ID attributes prohibit the use of the colon, thus creating a clash for anyone wishing to adopt URNs for universal naming, but also wishing to (reasonably) then use that universal name as the unique value in an XML document within an ID attribute. We are not alone in having to contend with this situation (recent e-mail exchange with Sun Microsystems expert on the ebXML project).

**Conclusion:** We will store the colon as an underscore `_' in our XML, and expect to rewrite to colons in any (non-XML) output situation where actual URNs are required.

**Additional note:** we have adopted a more forward-thinking approach to URN domain naming, simply using the same method familiar to Java class naming, that of placing the highest hierarchical domain to the left-most, and descending from there (e.g. edu.mit.ocw). This is in contrast to much of the URN naming encountered in these still early days of URN adoption (e.g. other organizational units at MIT might, for example, simply declare their names at the “top” of their URN, as in if we were to just lay claim to the three letter acronym “ocw” worldwide, with no context of what larger domain we operate in (mit) within (edu)).

### Organization Identifier Scheme

**Object Model Node**: `/manifest/organizations/organization/@identifier`

| Organization (OCW) | [orgTypePrefix value="ocw"] + "" + [Channel Name] | ocw_21H-418Technologies-of-Word-1450-2000Fall2002 |

### Item Identifier Scheme

**Object Model Node**: `/manifest/organizations/organization/item/@identifier`

<p>| Item (Section) | [itemTypePrefix value=&quot;i&quot;] + [Channel Name] + &quot;.&quot; + [Section Directory Name] | i21H-418Technologies-of-word-1450-2000Fall2002.Calendar |
| Item (Detail HTML) | [itemTypePrefix value=&quot;i&quot;] + [Channel Name] + &quot;.&quot; + [Section Directory Name] + &quot;.&quot; + [Detail HTML File Name] + &quot;.&quot; + [Detail HTML File Extension] | i21H-418Technologies-of-word-1450-2000Fall2002.Calendar.hausmanchapbooks.htm |
| Item (Detail HTML) | [itemTypePrefix value=&quot;i&quot;] + [Channel Name] + &quot;.&quot; + [Section Directory Name] + &quot;.&quot; + [Detail HTML File Name] + &quot;.&quot; + [Detail HTML File Extension] | i11-001JFall2001.LectureNotes.lec1.htm |</p>
<table>
<thead>
<tr>
<th>Item (Detail HTML) (Resource)</th>
<th>[itemTypePrefix value=&quot;i&quot;] + [Channel Name] + &quot;.&quot; + [Section Directory Name] + &quot;.&quot; + [Detail HTML File Name] + &quot;.&quot; + [Resource File Name] + &quot;.&quot; + [Resource File Extension]</th>
</tr>
</thead>
<tbody>
<tr>
<td>i11-001JFall2001.LectureNotes.lec1.01002t.jpg</td>
<td></td>
</tr>
<tr>
<td>i11-001JFall2001.LectureNotes.lec1.image1.html</td>
<td></td>
</tr>
<tr>
<td>i11-001JFall2001.LectureNotes.lec1.image1.01002.jpg</td>
<td></td>
</tr>
</tbody>
</table>

In the footnotes that follow, full URLs are shown, using **bold** to highlight the leaf string to be used in constructing the item/@identifier.

1 **Section HTML or Detail HTML**

A. For all "index.htm[l]" files, do not use File Name and Extension, instead use name of immediate parent directory.

   Section
   

   Detail HTML
   
   [INVENTED URLs]:
   
   

B. If not "index.htm[l]", then use File Name and Extension.

   Detail HTML
   
   [Real URLs]:
   
   

2 **Resource (linked to off Section HTML page)**

   Resource:
   

3 **Resource (linked to off Detail HTML page)**

   **Note:** This example OCW Resource (thumbnail image) is a Networked Resource ("NR"), not on Akamai Net Storage (ANS) (as the full-scale JPG is).

   Detail HTML:
   
   
   (links to…)

   Resource:
4 Detail HTML page (linked to off top-level Detail HTML page)

Note: This example OCW nested Detail HTML page we found is actually stored on Akamai Net Storage (ANS), and its HTML file will not be part of the content package. But this still serves as an example for the logic used in constructing the item @identifier for nested Detail HTML pages, wherever they might be stored (OcwWeb; NR).

Detail HTML: (top-level)

(links to…)

Detail HTML (nested):
http://ocw.mit.edu/ans7870/11/11.001j/f01/lectureimages/1/image1.html

5 Resource (linked to off nested Detail HTML page, linked to off top-level Detail HTML page)

Note: Here in version 1.0 of the CWSpace IMS-CP Profile, this example OCW Resource will not be part of the manifest nor PIF collection of files, as the OCW Resources on Akamai Net Storage (ANS) will not be part of the content package. This note therefore is only provided for completeness concerning the naming scheme.

Detail HTML: (top-level)

(links to…)

Detail HTML (nested):
http://ocw.mit.edu/ans7870/11/11.001j/f01/lectureimages/1/image1.html

(links to…)

Resource:
http://ocw.mit.edu/ans7870/11/11.001j/f01/lectureimages/1/01002.JPG

Item Identifier Prefix Vocabularies

<table>
<thead>
<tr>
<th>Identifier Element</th>
<th>Value List</th>
<th>Value Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrgTypePrefix</td>
<td>ocw</td>
<td>An OCW course package</td>
</tr>
<tr>
<td></td>
<td>lo</td>
<td>An organization of Learning Objects</td>
</tr>
<tr>
<td></td>
<td>pack</td>
<td>An organization of prior versions of this package or manifest</td>
</tr>
<tr>
<td>ItemTypePrefix</td>
<td>i</td>
<td>An item in an OCW course organization</td>
</tr>
<tr>
<td></td>
<td>l</td>
<td>An item in an organization of Learning Objects</td>
</tr>
<tr>
<td></td>
<td>p</td>
<td>An item in an organization of manifests</td>
</tr>
</tbody>
</table>

Resource Identifier Scheme
/manifest/resources/resource/@identifier

Please Note: Here the term “Resource” refers to the IMS-CP term, not the OCW term.
Resource (Items)  
file path from root, including complete file name and extension  
- exclude the leading slash  
- substitute "." for "/" throughout  

NR rdonlyres.Global.0.0AE311FE-4824-488F-A587-520C34EF24D2.0.chp_erasmus.imac1.jpg  
an7870.21h.21h.418.philip.index.html  

Resource (non Items)  
"webrenderrequirements" + "." + [channel name]  

webrenderrequirements.21H-418Technologies-of-Word-1450-2000Fall2002  

Please Note: Here in version 1.0 of this profile, files for OCW Resources stored at ANS (Akamai) will not be stored in the Content Package. Therefore, the IMS-CP <resource> element will use the @href attribute to contain a complete URL to the ANS file:

<resource identifier="ans7870.21h.21h.418.philip.index.html" type="webcontent" href="http://ocw.mit.edu/ans7870/21h/21h.418/philip/index.html"/>

As seen, an @identifier is still created, according to the rules and examples set out above.

**Standard Schemas**

The standard, publicly available schemas referred to in the IMS Content Package specification are listed below. Current versions are noted in the filename or path directory.

One point of discussion regarding the schemas is whether it is of critical importance to package the schema files locally, in the root of the package, and to write the XML instance documents to point to the schema there.

This IMS-CP Profile for CWSpace maintains the position that web-aware schema detection is not an unreasonable stance to take for the use cases for the content packages that will be exchanged among online learning and repository systems. Desktop tools for the authoring or editing of content packages are less likely to insist on local copies of schema only (whereas “run” or play environments, usually more SCORM compliant than IMS-CP tools, may be far more likely to require local access to schemas). But our use cases don’t foresee run/play usage within desktop tools; the desktop serves as an authoring environment for materials that will be uploaded and hosted and “played” from online, web-aware LMSs and LORs.

**IMS-CP**

http://www.imsglobal.org/xsd/imscp_v1p1.xsd

**IEEE LOM**

http://ltsc.ieee.org/xsd/lomv1.0/lom.xsd

**ADL SCORM CP**

http://www.adlnet.org/xsd/adlcp_rootv1p2.xsd
**Extension Schemas**

The extension schema files are to be maintained on servers controlled by the owning organization. At the moment of writing this document, the OCW IMS-CP extension schema is temporarily hosted on a CWSpace server, instead of the final location on an OCW server.

**cwspace_imscp.xsd**

http://cwspace.mit.edu/docs/xsd/cwspace_imscp/version10/cwspace_imscp.xsd

New elements, attributes: checksum, sourceSystem, profile, profileVersion, manifest/@packageType

**ocw_imscp.xsd**

http://cwspace.mit.edu/docs/xsd/ocw_imscp/version10/ocw_imscp.xsd

New element: sectionTemplateType

**Miscellaneous Topics**

**<item><title> Values**

The string value to be included in the <title> element for <item>s will ideally be the string that is found in the associated LOM metadata's title element.

Other options addressed included omitting the title (this is acceptable IMS-CP); using the “clickable text” off the linking web page (though a vast majority of these would read nothing more than “PDF”); and finally the actual filename (often cryptic though sometimes indicative of content).

**OCW Global Artifacts**

Interestingly, from the OCW point of view, as they prepare their course content for other sorts of purposes (professor’s course copies, translation partners, other), they view the “wrapper” “parent” set of pages as intrinsically useful for those contexts. These include the OCW Home Page; the relevant Department Page; the Course List Page.

For the CWSpace IMS-CP Profile however, these artifacts are not to be made part of the course content package, which needs a more independent existence when “shipped around” as an OCW course among the LMSs and the DSpace (LOR).

Please note: OCW Global artifacts that are used in course pages are to be part of the package! These include things like Logo.gif, spacer.gif, core.css, etc.