



# IMS Question and Test Interoperability Integration Guide

## Version 2.1 Public Draft Specification

Copyright © 2006 IMS Global Learning Consortium, Inc. All Rights Reserved.

The IMS Logo is a registered trademark of IMS GLC.

Document Name: IMS Question and Test Interoperability Integration Guide

---

Date Issued: 8 June 2006

Caution: this specification is incomplete in its current state. The IMS QTI project group is in the process of evolving this specification based on input from market participants. Suppliers of products and services are encouraged to participate by contacting Mark McKell at [mmckell@msglobal.org](mailto:mmckell@msglobal.org). This specification will be superseded by an updated release based on the input of the project group participants.

Please note that supplier's claims as to implementation of QTI v2.1 and conformance to it HAVE NOT BEEN VALIDATED by IMS GLC. While such suppliers are likely well-intentioned, IMS GLC member organizations have not yet put in place the testing process to validate these claims. IMS GLC currently grants a conformance mark to the Common Cartridge profile of QTI v1.2.1. The authoritative source of products and services that meet this conformance is contained in the IMS online product directory

<http://www.msglobal.org/ProductDirectory/directory.cfm>

Thank you for your interest in and support of IMS QTI.

### IPR and Distribution Notices

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: [http://www.msglobal.org/ipr/imsipr\\_policyFinal.pdf](http://www.msglobal.org/ipr/imsipr_policyFinal.pdf).

Copyright © 2006 IMS Global Learning Consortium. All Rights Reserved.

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

This document may be copied and furnished to others by Registered Users who have registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as

needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS project group.

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

The limited permissions granted above are perpetual and will not be revoked by IMS or its successors or assigns.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

# Table of Contents

1. [Introduction](#)
2. [References](#)
3. [Content Packaging](#)
  - 3.1. [Packaging Items](#)
  - 3.2. [Packaging Tests](#)
  - 3.3. [Packaging Item and Test Fragments](#)
  - 3.4. [Packaging Response Processing Templates](#)
  - 3.5. [Associating Meta-data to items in a Content Package](#)
  - 3.6. [Packaging QTI Version 1 Objects](#)
4. [Learning Design](#)
  - 4.1. [Integration through IMS LD Properties and QTI Variables](#)
    - 4.1.1. [Naming & Typing](#)
  - 4.2. [How and where QTI content is referenced in the Learning Design of a Unit of Learning](#)
  - 4.3. [A skeletal example of IMS LD, QTI and CP integration](#)
  - 4.4. [IMS LD and QTI assessmentTest integration](#)
5. [Simple Sequencing](#)
6. [CMI](#)

# 1. Introduction

This document describes a set of methods for using QTI's [assessmentTests](#) or processing templates in combination with some other commonly used technical specifications.

## 2. References

### APIS

Assessment Provision through Interoperable Segments

*Barr, Sclater and Young*

### ASI\_BIND

IMS Question & Test Interoperability: ASI XML Binding Specification, Version 1.2

Published: 2002-02

### CMI

IEEE 1484.11.1, Standard for Learning Technology - Data Model for Content Object Communication

### IMS\_CP

IMS Content Packaging Specification, Version 1.1.3

### IMS\_LD

IMS Learning Design Specification, Version 1.0

Published: 2003-01

### IMS\_MD\_Binding

IMS Learning Resource Meta-Data XML Binding, Version 1.2.1

### IMS\_SS

IMS Simple Sequencing Specification, Version 1.0

Published: 2003-03

### LOM

IEEE 1484.12.1-2002 Standard for Learning Object Meta-data (LOM)

### XHTML

XHTML 1.1: The Extensible HyperText Markup Language

### 3. Content Packaging

IMS Content Packaging [\[IMS\\_CP\]](#) should be used when transferring [assessmentItems](#), [assessmentTests](#), or processing templates between systems. To avoid confusion between the identically named *item* in the Content Packaging specification and, more generally, to make it clear when terms in this Integration Guide are referring to elements in the content packaging model, all references to these elements will be qualified with the prefix "cp:". This is purely a typographical convention and *does not* indicate the use of XML namespacing syntax.

In preparing this specification, every effort has been taken to ensure that no modifications or extensions to the existing Content Packaging data model are defined and, furthermore, features of that specification are used in the way originally intended. The goal is to enable the use of content packages containing assessment objects with the existing base of tools that support IMS Content Packaging without modification.

During the development of this version of the QTI specification, IMS has started the development of version 1.2 of IMS Content Packaging [\[IMS\\_CP\]](#). Though that version hasn't reached its final status yet, the QTI specification has been aligned to the expected changes in the IMS Content Packaging [\[IMS\\_CP\]](#) specification. Most notably is the change in the recommendation of the use of cp:schema of "IMS QTI Item" and cp:schemaversion of "2.0" for resources. This has now been deprecated following the expected change in IMS Content Packaging [\[IMS\\_CP\]](#) version 1.2.

Version 1 of this specification defined an *objectbank* as a collection of items and sections. In this version of the specification, this concept has been extended to include assessment tests. A collection of assessment objects in a content package is therefore considered to be an objectbank and the meta-data associated with the package as a whole is interpreted as the meta-data describing the objectbank. In other words, objectbanks are bound to content packages for interchange. Note that an item pool is simply treated as a special case of an objectbank. Packaged items may still be referred to individually in an associated learning design or set of sequencing rules. This type of integration is discussed later in this document.

Given that users may wish to package assessment content defined using version 1 of the QTI specification additional guidance is given on how to package *questestinterop* objects. These objects may define items, sections, assessments, or objectbanks in any way allowed by that version of this specification.

#### 3.1. Packaging Items

An IMS content package is a logical directory containing the content files and a special manifest file which describes them. An [assessmentItem](#) is represented in a content package by an XML file that satisfies the XML binding described by this specification.

For example, in the case of a single item the content package will contain:

1. the manifest (an XML file called `imsmanifest.xml`)
2. the item (a QTI XML file)
3. any auxiliary files (typically images or media files) required by the item

The manifest file must contain a **separate** `cp:resource` describing each item. The `cp:type` of the `cp:resource` must be `imsqti_item_xmlv2p1`. The `cp:resource` in turn must contain a `cp:file` representing the item's XML file. The `cp:resource` should also contain a `cp:file` for each of the item's auxiliary files, however, if an auxiliary file is shared amongst several items (whether or not these other items are contained in the same content package) the auxiliary file may be represented by a separate `cp:resource`. In this case, the item's `cp:resource` must contain a `cp:dependency` to the `cp:resource` representing the auxiliary file.

Meta-data may be associated with an item by including it in the `cp:resource`. In the case of a content package that contains only one item the meta-data, if given, **must** be included in the `cp:resource` and *not* the enclosing `cp:manifest`. Meta-data associated with the `cp:manifest` itself is reserved for describing the *package* (objectbank), not the package's contents.

The meta-data associated with an item should conform to the model and XML binding for item meta-data described elsewhere in this specification. Meta-data that conforms to that model must specify a *metadatascheme* of "QTIv2.1". The use of `cp:schema` of "IMS QTI Item" and `cp:schemaversion` of "2.0" is deprecated following the changes to the IMS Content Packaging [\[IMS CP\]](#) specification.

The `cp:type` of the `cp:resource` must be `imsqti_item_xmlv2p0` or `imsqti_item_xmlv2p1` depending on what version of the specification the item conforms to.

The use of a `cp:organization` to organize QTI items contained in a content package is reserved, except where the `cp:organization` is the basis for a set of rules described with the Simple Sequencing specification [\[IMS SS\]](#). Systems dealing only with the interchange of assessment objects conforming to this specification should ignore organizations when importing content packages.

Auxiliary files that are referred to using relative URIs in an [img](#) or [object](#) must be included in the content package. These files may be placed in the top level directory or in a sub-directory of the content package if preferred.

## 3.2. Packaging Tests

When a package not only contains items, but also a Test, the manifest file must contain a `cp:resource` describing the [assessmentTest](#). The `cp:type` of the `cp:resource` for the test must be `imsqti_test_xmlv2p1`. The `cp:resource` in turn must contain a `cp:file` representing the test's XML file. The `cp:resource` should also contain a `cp:file` for each of the test's auxiliary files. A package can contain multiple tests, each represented by its own `cp:resource`.

As with packages containing only items, the Items should be represented by a separate `cp:resource`. The relationship between items and test should be represented by a *cp:dependency* for each individual item in the `cp:resource` of the test.

A single [assessmentTest](#) can reference both items conforming to version 2.0 and items conforming to version 2.1 of this specification.

If an auxiliary file is shared with other tests or even other testitems (whether or not these are contained in the same content package) the auxiliary file may be represented by a separate `cp:resource`. In this case, the tests's `cp:resource` must contain a *cp:dependency* to the `cp:resource` representing the auxiliary file.

Meta-data specific to the test may be associated with a test by including it in the `cp:resource`. If given, this meta-data **must** be included in the `cp:resource` and *not* the enclosing `cp:manifest`. Meta-data associated with the `cp:manifest` itself is reserved for describing the *package*, not the package's contents.

The use of a `cp:organization` to organize QTI Tests contained in a content package is reserved for future use, except where the `cp:organization` is the basis for a set of rules described with the Simple Sequencing specification [\[IMS\\_SS\]](#).

Auxiliary files that are referred to using relative URIs in an [img](#) or [object](#) must be included in the content package. These files may be placed in the top level directory or in a sub-directory of the content package if preferred.

### 3.3. Packaging Item and Test Fragments

An item fragment is part of an item that is managed independently of the items that depend on it. Similarly, a test fragment is part of a test that is managed independently of the tests that depend on it. The `cp:type` of the `cp:resource` for the test must be `imsqti_fragment_xmlv2p1`. The `cp:resource` in turn must contain a `cp:file` representing the fragments XML file. A package can contain multiple fragments, each represented by its own `cp:resource`.

The relationship between items and fragments *should* be represented by a *cp:dependency* for the fragment(s) in the `cp:resource` of the item that uses the template.

### 3.4. Packaging Response Processing Templates

There usually is no real need to include the standard templates in a content package, but they can be included in the package containing items. The package can also contain custom response processing templates shared by the items in the package. The `cp:type` of the `cp:resource` for the test must be `imsqti_rptemplate_xmlv2p1`. The `cp:resource` in turn must contain a `cp:file` representing the template's XML file. A package can contain multiple templates, each represented by its own `cp:resource`.



The relationship between items and templates should be represented by a *cp:dependency* for the template in the *cp:resource* of the item that uses the template.

### 3.5. Associating Meta-data to items in a Content Package

Previous versions of the IMS QTI specification had a specific meta-data set contained within the data structures themselves, i.e. the ASI. That meta-data vocabulary had its own set of names, all of which started with the characters 'qmd\_'. The [Migration Guide](#) document describes how to convert these elements for use in QTI version 2.1.

In QTI version 2.0, QTI-specific meta-data has been brought into line with the IEEE LOM in accordance with the IMS Meta-data Best Practice and Implementation Guide for [\[LOM\]](#). The IEEE LOM standard defines a set of meta-data elements that can be used to describe learning resources, but does not describe assessment resources in sufficient detail. The application profile provided in this document therefore extends the IEEE LOM to meet the specific needs of QTI developers wishing to associate meta-data with items (as defined by the accompanying Item Information Model).

The [Meta-data and Usage Data](#) document describes a profile of [\[LOM\]](#) suitable for use with assessment items and a separate data model for representing usage data (i.e., item statistics). This document will be of particular interest to developers and managers of item banks and other content repositories, and to those who construct assessments from item banks

Meta-data may be associated with an item by including it in the *cp:resource*. In the case of a content package that contains only one item the meta-data, if given, **must** be included in the *cp:resource* and *not* the enclosing *cp:manifest*. Meta-data associated with the *cp:manifest* itself is reserved for describing the *package*, not the package's contents.

One of the goals during the development of this specification was to allow support by as large a number of existing tools as possible. It was also considered important to provide support for use of both the QTI specific meta-data in combination with the currently widely used [\[IMS MD Binding\]](#) and the future IEEE [\[LOM\]](#) binding. Both of these have been taken into account while developing the integration method.

The complete example file contains all the code discussed below:

#### Meta-data in Content Package example [examples/mdexample/imsmanifest.xml](#)

The schema file for the IMS QTI Version 2.1 meta-data needs to be referenced in the manifest element using a suggested prefix of **imsqti**.

```
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
  xmlns:imsqti="http://www.imsglobal.org/xsd/imsqti_v2p1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  identifier="MANIFEST-QTI-1"
```

```
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd
http://www.imsglobal.org/xsd/imsqti_v2p1 imsqti_v2p1.xsd">
```

In this case the XSD-files **should** be included in the content package. Alternatively the schema files on the IMS website can be referenced instead of including them in the package.

```
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:imsqti="http://www.imsglobal.org/xsd/imsqti_v2p1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
identifier="MANIFEST-QTI-1"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
http://www.imsglobal.org/xsd/imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2
http://www.imsglobal.org/xsd/imsmd_v1p2p2.xsd
http://www.imsglobal.org/xsd/imsqti_v2p1
http://www.imsglobal.org/xsd/imsqti_v2p1.xsd">
```

The meta-data for the individual items is added inside the meta-data element for the resource element containing the QTI item. The generic meta-data, either using the IMS MD structures or the IEEE LOM structures goes first in the cp:metadata element, followed by the IMS QTIv2 Item meta-data.

```
<resource identifier="RES-1" type="imsqti_item_xmlv2p1"
href="qti_v2_item_01.xml">
  <metadata>
    <imsmd:lom>
      <imsmd:general>
        <imsmd:identifier>qti_v2_item_01</imsmd:identifier>
        <imsmd:title>
          <imsmd:langstring xml:lang="en">Meta-data Example Item #1</imsmd:langstring>
        </imsmd:title>
        <imsmd:description>
          <imsmd:langstring xml:lang="en">This is a dummy item</imsmd:langstring>
        </imsmd:description>
      </imsmd:general>
      <imsmd:lifecycle>
        <imsmd:version>
          <imsmd:langstring xml:lang="en">1.0.1</imsmd:langstring>
        </imsmd:version>
        <imsmd:status>
          <imsmd:source>
            <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
          </imsmd:source>
          <imsmd:value>
            <imsmd:langstring xml:lang="x-none">Draft</imsmd:langstring>
          </imsmd:value>
        </imsmd:status>
      </imsmd:lifecycle>
      <imsmd:metametadata>
        <imsmd:metadatascheme>LOMv1.0</imsmd:metadatascheme>
        <imsmd:metadatascheme>QTIv2.1</imsmd:metadatascheme>
        <imsmd:language>en</imsmd:language>
      </imsmd:metametadata>
    </imsmd:lom>
  </metadata>
```

```

</imsmd:metametadata>
<imsmd:technical>
<imsmd:format>text/x-imsqti-item-xml</imsmd:format>
<imsmd:format>image/png</imsmd:format>
</imsmd:technical>
</imsmd:lom>
<imsqti:qtiMetadata>
<imsqti:timeDependent>>false</imsqti:timeDependent>
<imsqti:interactionType>choiceInteraction</imsqti:interactionType>
<imsqti:feedbackType>nonadaptive</imsqti:feedbackType>
<imsqti:solutionAvailable>>true</imsqti:solutionAvailable>
<imsqti:toolName>XMLSPY</imsqti:toolName>
<imsqti:toolVersion>5.4</imsqti:toolVersion>
<imsqti:toolVendor>ALTOVA</imsqti:toolVendor>
</imsqti:qtiMetadata>
</metadata>
<file href="choice.xml"/>
<file href="images/sign.png"/>
</resource>

```

The above code shows a simple example containing both meta-data from the IMS meta-data specification, describing the title, description, version and status of the item. The second part of the meta-data is IMS QTI specific. It is not mandatory to use both meta-data parts, either the generic IMS meta-data part or the QTI specific part of the meta-data may be used alone.

### 3.6. Packaging QTI Version 1 Objects

In QTI version 1, the *questestinterop* element was used to contain individual items, sections, assessments, or object banks. All of these objects, including any collections allowed by that version of the specification are bound to XML documents of the same type, documents with a top level <questestinterop> element.

When packaging questestinterop instances, the content package will contain:

1. the manifest
2. the questestinterop file (a QTI XML file satisfying the binding described in

[\[ASI BIND\]](#))

3. any auxiliary files (typically images or media files) required by the questestinterop object

The manifest file must contain a **separate** cp:resource describing each questestinterop object. The cp:type of the cp:resource must be `imsqti_questestinterop_xmlv1p2`. The cp:resource in turn must contain a cp:file representing the questestinterop's XML file. The cp:resource should also contain a cp:file for each of the auxiliary files or, alternatively, a cp:dependency to a separate cp:resource representing the auxiliary file. The two approaches may be mixed in the same content package and must be treated identically.

Meta-data may be associated with a questestinterop object in the cp:manifest by adding it to the associated cp:resource. Meta-data conforming to the model and XML binding for item meta-data

described elsewhere in this specification must only be used when the `questestinterop` object contains a single item and no section, assessment or objectbank objects. Furthermore, this item *must not* contain either *qtimetadata* or *itemmetadata* elements.

The use of a `cp:organization` to impose a hierarchical ordering on QTI `questestinterop` objects contained in a content package is forbidden.

## 4. Learning Design

IMS Learning Design [\[IMS LD\]](#) and IMS QTI are natural partners in the learning process. The central objective of the IMSLD specification is to model Units of Learning, delimited pieces of education or training, such as courses, modules, lessons, etc. A unit of learning describes the teaching-learning process and includes a variety of activities, assessments, services and support facilities provided by teachers, trainers, and other staff members.

The primary motivation for integrating IMS Learning Design (IMSLD) and QTI stems from use cases which involve exploiting the results of a test or assessment to influence the learning process, often referred to as formative assessment. However, other use cases involving summative assessment, a final and comprehensive test of the learner's level of understanding, also form part of the rationale for integration.

A typical learning situation involving a Unit of Learning containing a test or assessment might see learners engaging in a series of learning activities, then undergoing a short assessment. On the basis of the results of this assessment, learners experiencing difficulties with material are directed to additional learning activities designed to strengthen their understanding. In addition, those learners scoring particularly well might be directed to skip certain learning activities.

Other cases of integration exist, such as the incorporation of straightforward, low-threshold questions throughout a unit of learning to maintain a certain degree of learner interaction, and the use of assessments for group formation in group-based learning (e.g., when dividing learners into groups of similar levels of ability).

In general, the integration seeks to position assessment in its wider educational context, and revolves around the results of *QTI-based processes* being used in *IMSLD-based processes*. However, communication is not restricted to one direction; information can be brought in to influence the assessment process, including learner preferences, the results of previous tests, or the time taken to reach the assessment.

Indeed, reconciling information described using IMSLD with that described using QTI so that run-time systems interoperate appropriately is a particular case of more general inter-specification interoperability. This Integration Guide does not seek to solve the more general issue, but aims to foster a tighter integration of IMSLD and QTI to help extend their application areas and improve the benefit of their combined use.

### 4.1. Integration through IMS LD Properties and QTI Variables

A large part of IMS LD's flexibility in orchestrating learning flows comes from the use of properties and conditions. Properties are used to record various types of information, which can subsequently be examined and used to influence aspects of the learning process, including the ordering and visibility of learning activities and learning objects. IMS LD supports the definition of different types of property in terms of scope and persistence (see the IMS Learning Design Specification for details). Note that the use of properties implies units of learning at IMS LD level B or C.

In a typical example, learning designers might create a property called P-intake-test-result:

```
<imsld:properties>
  <imsld:locpers-property identifier="P-intake-test-result">
    <imsld:title>The result for the test carried out as the first step in
a learning flow</imsld:title>
    <imsld:datatype datatype="integer"/>
    <imsld:initial-value>0</imsld:initial-value>
  </imsld:locpers-property>
</imsld:properties>
```

Here we see that an IMSLD (local, personal) property is declared, with type integer and with an initial value of zero. This property might be used in a condition to select between alternative learning activities, based on the results of the test:

```
<imsld:conditions>
  <imsld:if>
    <imsld:less-than>
      <imsld:property-ref ref="P-intake-test-result"/>
      <imsld:property-value>3</imsld:property-value>
    </imsld:less-than>
  </imsld:if>
  <imsld:then>
    <imsld:show>
      <imsld:learning-activity-ref ref="LA-Review-Additional-Material"/>
    </imsld:show>
  </imsld:then>
</imsld:conditions>
```

Here we see an IMSLD condition which states that if the value of P-intake-test-result is less than 3, a learning activity should be shown (giving, for example, additional guidance on a topic).

The IMS QTI specification includes the similar notion of variables, used to record the outcomes of assessment:

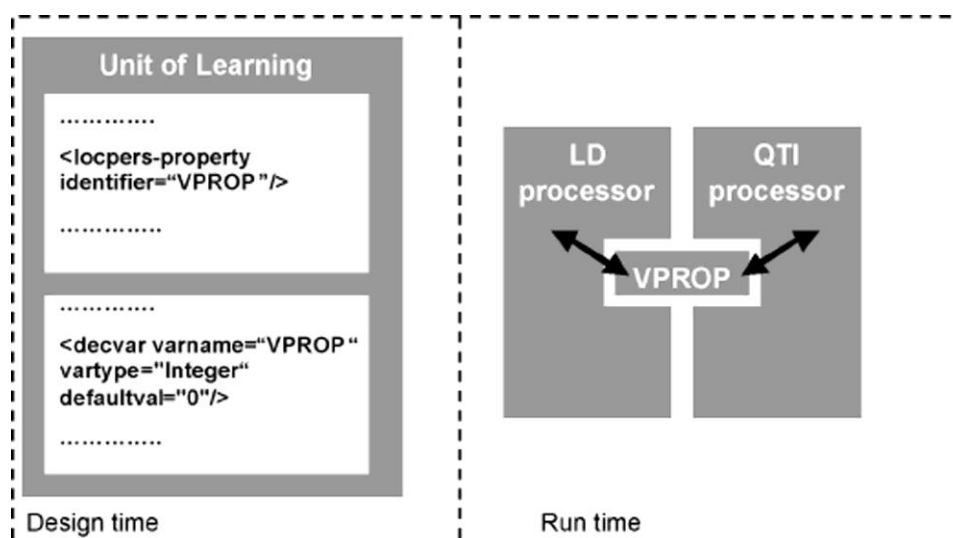
```
<imsqti:outcomes_processing>
  <imsqti:outcomes>
    <imsqti:decvar varname="SCORE" vartype="Integer" defaultval="0"/>
  </imsqti:outcomes>
</imsqti:outcomes_processing>
```

The above XML fragment shows the declaration of a variable SCORE, of type integer and with an initial value of zero.

The integration of IMS LD and IMS QTI revolves around aligning property and variable names. Essentially, when property identifiers and variable names are declared to be lexically identical at design time (i.e. in IMS LD-based and IMS QTI-based XML), they are considered to be a *shared variable* in run-time software environments which involve IMS LD and IMS QTI-based processes. To achieve this effect in the above QTI example, the variable name would need to be modified to become the property name used in IMSLD XML. Alternatively, the IMSLD XML

could be modified to align it with the QTI variable name (including references in any IMSLD conditions using the property).

Figure 4.1 shows a Unit of Learning which includes both a fragment of QTI and a fragment of IMS LD which refer to VPROP in the manner describe above. These fragments may have been created with different tools or using an editing environment which supports authors in creating both types of content. Although software may be written to interpret both specifications, effectively side-stepping the integration issue, separate markets of LD processors and QTI processors already exist. As a result, the run-time situation may well involve different processors reading and writing VPROP



**Figure 4.1 Illustration of LD and QTI integration.**

This is a loose level of integration in that, from the perspective of IMS LD, the internal response processing algorithms used in the QTI content are hidden, with only the resulting outcome being of importance. Similarly, QTI-based processes are unaware of any IMS LD-based use of outcomes. Some run-time mechanism must be in place to enable both IMS LD and IMS QTI-based processes to *write to* and *read from*, and services based on the emerging IMS Shareable State Persistence Information Model would appear a suitable candidate.

A complicating factor with this approach lies in the use of multiple QTI items in which the same QTI variable name may be used more than once. The QTI specification indicates the default variable name to be "SCORE", and it is not uncommon to see this variable name used with QTI items. In order to avoid naming clashes and increase the transparency of Units of Learning which integrate IMSLD and IMS-QTI, the recommended best practice is to combine identifiers. The approach is to create compound identifiers for use as IMS LD property names by combining the resource identifier associated with the content package resource containing the IMS QTI item as a prefix to the variable name, using a period as separator. This approach is illustrated below.

#### 4.1.1. Naming & Typing

LD's properties are of type XML Identifiers (see [http://www.imsglobal.org/learningdesign/ldv1p0/imsld\\_info1p0.html#1515694](http://www.imsglobal.org/learningdesign/ldv1p0/imsld_info1p0.html#1515694)), with the rules for their construction being governed by the XML 1.0 specification:

*Definition: A Name is a token beginning with a letter or one of a few punctuation characters, and continuing with letters, digits, hyphens, underscores, colons, or full stops, together known as name characters.] (from Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 <http://www.w3.org/TR/2004/REC-xml-20040204/#NT-Nmtokens>)*

QTIv2 item variables are also XML Identifiers and so the rules governing the lexical composition of IMSLD properties and IMSQTI variable are identical.

However, the type systems used in IMS LD and IMS QTI differ:

Learning Design	QTI
<i>no equivalent</i>	identifier
Boolean	boolean
Integer	integer
Real	float
String	string
Text	string
<i>no equivalent</i>	point
<i>no equivalent</i>	pair
<i>no equivalent</i>	directedPair
Duration	duration
File	file
URI	uri
Datetime	<i>no equivalent</i>

A final complicating factor is the presence of multi-valued variables in QTI which have no equivalent in IMS LD.

These issues point to the need for systems which process QTI items in combination with IMS LD to perform checks and issue warnings when differences are seen either in the type or multi-valued nature of QTI outcome variables and IMS LD properties which share a local part of their QName.

## 4.2. How and Where QTI Content is Referenced in the Learning Design of a Unit of Learning

Conceptually, from an IMS Learning Design perspective, tests are linked to learning-activities which provide the instruction to complete the test that is present in the environment. In this way,



reference to QTI items is likely to be as learning-objects in environments associated with learning-activities. The following figure shows part of an IMS manifest file containing these relationships:

### **Screenshot illustrating a Learning Design referring to a QTI Item**

[images/ldManifest.png](#)

A learning activity can reference one or more environments, and nesting of environments is permitted. Each environment can, in turn, contain several learning-objects, each referencing a separate QTI item. Environments can be made visible or hidden using IMS Learning Design's condition mechanism, giving the designer of a unit of learning considerable flexibility in modeling the sequencing and selection of test items during the course of a learning process. Note that the same approach can be used for including QTI content above the item level (sections or assessments).

Compound identifiers are formed by using the resource identifier of the content package resource which references the QTI item, together with the QTI variable name, separated by a period. In the above example, assuming the QTI content in the file `choice_01.xml` contains an outcome variable declaration for the variable `SCORE`, IMS-LD content needing to use the value of `SCORE` would use a property declared as:

```
<imsld:locpers-property identifier="Question1.SCORE">
```

## **4.3. A skeletal example of IMS LD, QTI, and CP integration**

This section walks the reader through a skeletal example of a Unit of Learning which integrates IMS Learning Design and IMS QTI in a content package. Three questions are used, two of which use the same variable names in order to further illustrate the approach to compound identifiers.

We will build the example starting from the QTI content, held in three separate files:

**choice\_01.xml**

[examples/ldexample/choice\\_01.xml](#)

**choice\_02.xml**

[examples/ldexample/choice\\_02.xml](#)

**choice\_03.xml**

[examples/ldexample/choice\\_03.xml](#)

Note that the first two items both use the variable `SCORE`, the third item uses the variable `SIGNSCORE`.

These three IMS QTI items would be included in the content package as three resources (note that two other resources are also shown in the resources section):

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Colin
Tattersall (Open University of the Netherlands) -->
<imscp:manifest xmlns:imscp="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsld="http://www.imsglobal.org/xsd/imsld_v1p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
  http://www.imsglobal.org/xsd/imsld_v1p0 IMS_LD_Level_B.xsd"
  identifier="Integration-Example">
  <imscp:organizations>
    <imsld:learning-design identifier="LD-Integration-Example" uri=""
level="B">
      ...
    </imsld:learning-design>
  </imscp:organizations>
  <imscp:resources>
    <imscp:resource identifier="R-Simple" type="webcontent">
      <imscp:file href="simple.xml"/>
    </imscp:resource>
    <imscp:resource identifier="Question_1" type="imsqti_item_xmlv2p1"
href="choice_01.xml">
      <imscp:file href="choice_01.xml"/>
    <imscp:file href="sign.png"/>
    </imscp:resource>
    <imscp:resource identifier="Question_2" type="imsqti_item_xmlv2p1"
href="choice_02.xml">
      <imscp:file href="choice_02.xml"/>
    <imscp:file href="sign2.png"/>
    </imscp:resource>
    <imscp:resource identifier="Question_3" type="imsqti_item_xmlv2p1"
href="choice_03.xml">
      <imscp:file href="choice_03.xml"/>
    <imscp:file href="sign3.png"/>
    </imscp:resource>
    <imscp:resource identifier="R-Feedback" type="webcontent">
      <imscp:file href="feedback.xml"/>
    </imscp:resource>
  </imscp:resources>
</imscp:manifest>

```

In this example, the three resources are referenced from a single environment associated with a single learning activity:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Colin
Tattersall (Open University of the Netherlands) -->
<imscp:manifest xmlns:imscp="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsld="http://www.imsglobal.org/xsd/imsld_v1p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
  http://www.imsglobal.org/xsd/imsld_v1p0 IMS_LD_Level_B.xsd"
  identifier="Integration-Example">
  <imscp:organizations>
    <imsld:learning-design identifier="LD-Integration-Example" uri=""
level="B">

```

```

...
<imsld:activities>
<imsld:learning-activity isvisible="true" identifier="LA-Signpost">
  <imsld:title>Complete the question show in the
environment</imsld:title>
  <imsld:environment-ref ref="E-Simple-Environment"/>
  <imsld:activity-description>
<imsld:title>Check your understanding of signposts</imsld:title>
<imsld:item identifier="I-Simple" identifierref="R-Simple"/>
</imsld:activity-description>
...
</imsld:learning-activity>
</imsld:activities>
<imsld:environments>
<imsld:environment identifier="E-Simple-Environment">
<imsld:title>Quick Test</imsld:title>
<imsld:learning-object identifier="LO-QTI-Item1">
<imsld:title>Assign a sign</imsld:title>
<imsld:item identifier="I-Question1" identifierref="Question_1"/>
</imsld:learning-object>
<imsld:learning-object identifier="LO-QTI-Item2">
<imsld:title>Assign a second sign</imsld:title>
<imsld:item identifier="I-Question2" identifierref="Question_2"/>
</imsld:learning-object>
<imsld:learning-object identifier="LO-QTI-Item3">
<imsld:title>And try to assign a third one</imsld:title>
<imsld:item identifier="I-Question3" identifierref="Question_3"/>
</imsld:learning-object>
</imsld:environment>
</imsld:environments>
</imsld:components>
<imsld:method>
...
</imsld:method>
</imsld:learning-design>
</imscp:organizations>
<imscp:resources>
<imscp:resource identifier="R-Simple" type="webcontent">
<imscp:file href="simple.xml"/>
</imscp:resource>
<imscp:resource identifier="Question_1" type="imsqti_item_xmlv2p0"
href="choice_01.xml">
  <imscp:file href="choice_01.xml"/>
  <imscp:file href="sign.png"/>
</imscp:resource>
<imscp:resource identifier="Question_2" type="imsqti_item_xmlv2p0"
href="choice_02.xml">
  <imscp:file href="choice_02.xml"/>
  <imscp:file href="sign.png2"/>
</imscp:resource>
<imscp:resource identifier="Question_3" type="imsqti_item_xmlv2p0"
href="choice_03.xml">
  <imscp:file href="choice_03.xml"/>
  <imscp:file href="sign3.png"/>
</imscp:resource>
<imscp:resource identifier="R-Feedback" type="webcontent">
<imscp:file href="feedback.xml"/>

```

```

        </imscp:resource>
    </imscp:resources>
</imscp:manifest>

```

IMS LD properties are defined for each outcome variable used in each resource file in the following manner, together with a property which will be used to hold the sum of the three:

```

<imsld:properties>
  <imsld:locpers-property identifier="Question_1.SCORE">
    <imsld:title>The result for the first question</imsld:title>
    <imsld:datatype datatype="integer"/>
    <imsld:initial-value>0</imsld:initial-value>
  </imsld:locpers-property>
  <imsld:locpers-property identifier="Question_2.SCORE">
    <imsld:title>The result for the second question</imsld:title>
    <imsld:datatype datatype="integer"/>
    <imsld:initial-value>0</imsld:initial-value>
  </imsld:locpers-property>
  <imsld:locpers-property identifier="Question_3.SIGNSCORE">
    <imsld:title>The result for the third question</imsld:title>
    <imsld:datatype datatype="integer"/>
    <imsld:initial-value>0</imsld:initial-value>
  </imsld:locpers-property>
  <imsld:locpers-property identifier="Total">
    <imsld:title>The total</imsld:title>
    <imsld:datatype datatype="integer"/>
    <imsld:initial-value>0</imsld:initial-value>
  </imsld:locpers-property>
</imsld:properties>

```

Assuming a pedagogical approach of programmed instruction, the learner may only progress if the total score is three (i.e., if all three questions are answered correctly). This is handled in IMS LD by indicating that the activity can only be completed when a property is set (`when-property-value-is-set`):

```

<imsld:activities>
  <imsld:learning-activity isvisible="true" identifier="LA-Signpost">
    <imsld:title>Complete the question show in the
environment</imsld:title>
    <imsld:environment-ref ref="E-Simple-Environment"/>
    <imsld:activity-description>
      <imsld:title>Check your understanding of signposts</imsld:title>
      <imsld:item identifier="I-Simple" identifierref="R-Simple"/>
    </imsld:activity-description>
    <imsld:complete-activity>
      <imsld:when-property-value-is-set>
        <imsld:property-ref ref="Total"/>
        <imsld:property-value>3</imsld:property-value>
      </imsld:when-property-value-is-set>
    </imsld:complete-activity>
    <imsld:on-completion>
    <imsld:feedback-description>

```

```

        <imsld:item identifier="I-Feedback" identifierref="R-Feedback"/>
      </imsld:feedback-description>
    </imsld:on-completion>
  </imsld:learning-activity>
</imsld:activities>

```

The final step is to include an IMS LD condition which sets the property Total to have the value 3 when all three questions have been answered correctly:

```

<imsld:conditions>
  <imsld:if>
    <imsld:and>
      <imsld:greater-than>
        <imsld:property-ref ref="Question_1.SCORE"/>
        <imsld:property-value>0</imsld:property-value>
      </imsld:greater-than>
      <imsld:greater-than>
        <imsld:property-ref ref="Question_2.SCORE"/>
        <imsld:property-value>0</imsld:property-value>
      </imsld:greater-than>
      <imsld:greater-than>
        <imsld:property-ref ref="Question_3.SIGNSCORE"/>
        <imsld:property-value>0</imsld:property-value>
      </imsld:greater-than>
    </imsld:and>
  </imsld:if>
  <imsld:then>
    <imsld:change-property-value>
      <imsld:property-ref ref="Total"/>
      <imsld:property-value>3</imsld:property-value>
    </imsld:change-property-value>
  </imsld:then>
</imsld:conditions>

```

The property Total could equally have been given the type *boolean* and assigned a value of true once the total had been reached.

### Full Manifest File

<examples/ldexample/imsmanifest.xml>

## 4.4. IMS LD and QTI assessmentTest integration

The previous example shows how individual QTI [assessmentItem](#)s are referenced from within IMS Learning Design. In a similar way it is possible to point to complete [assessmentTests](#) from within IMS LD. The value of the outcome variable SCORE of the test would use a property declared as:

```
<imsld:locpers-property identifier="Assessment1.SCORE">
```

Assessment1 in that case is the identifier of the resource containing the QTI [assessmentTest](#).

## 5. Simple Sequencing

The IMS Simple Sequencing specification [\[IMS\\_SS\]](#) defines a method for representing the intended sequencing behavior of an authored learning experience. In other words, the way learning technology systems can sequence discrete learning activities in a consistent way.

Integrating individual assessment items with learning activities into a single activity tree controlled by simple sequencing rules suggests the need to map the information about the item session into the appropriate simple sequencing concepts. A detailed discussion of how this mapping might be implemented is beyond the scope of this document, though a number of experimental approaches are currently being discussed within the community. Readers may wish to monitor the output of the [\[APIS\]](#) project which is currently preparing more detailed recommendations.

To help facilitate easier integration of QTI items into activity trees authors are strongly encouraged to define an outcome variable with the name *SCORE* to be of a numeric type and to provide a value for the [normalMaximum](#) attribute of the [outcomeDeclaration](#). Note particularly that scores will be normalized to the range [-1.0,1.0] for use with sequencing rules so care will need to be taken when designing items that generate scores with minimum obtainable values other than 0.

Simple sequencing makes use of the organization element within the content package. In order to treat an assessment item as an activity a suitable cp:item element will need to be included in the organization. As stated above, this is the only use of the organization element that this specification recommends when packaging QTI content.

Note that assessment items have resource type `imsqti_item_xmlv2p0` and not `webcontent`. This indicates to the processing system that the resource is not suitable for handing directly to a web-browser for presentation. Instead, web-based systems that support the use of assessment items within activity trees must be capable of converting the assessment item into a suitable form for web delivery, processing the responses and collecting and mapping the resulting item outcomes. The version 2 data model for the [itemBody](#) profiles [\[XHTML\]](#) making this transformation considerably easier.

## 6. CMI

The CMI datamodel [\[CMI\]](#) is designed to provide a mechanism for the communication of data created at runtime, such as the outcomes of an assessment item, to a learning management system or other runtime service. A detailed description of how to map the contents of an item session into this model is beyond the scope of this specification.

Item authors are encouraged to follow the best practice described in [Simple Sequencing](#) to help facilitate interoperability through the CMI interface. A special predefined item variable (*completionStatus*) is defined to ease integration with CMI-based systems. Authors are reminded that adaptive items must maintain a value for this variable in the item's [responseProcessing](#) rules.

## About This Document

<b>Title</b>	IMS Question and Test Interoperability Integration Guide
<b>Editors</b>	Steve Lay (Cambridge Assessment), Pierre Gorissen (SURF)
<b>Version</b>	Public Draft v2.1
<b>Version Date</b>	8 June 2006
<b>Status</b>	<b>Public Draft Specification</b>
<b>Summary</b>	This document describes the QTI Integration Guide specification.
<b>Revision Information</b>	8 June 2006
<b>Purpose</b>	This document has been approved by the IMS Technical Board and is made available for public review and comment.
<b>Document Location</b>	

## List of Contributors

The following individuals contributed to the development of this document:

<b>Name</b>	<b>Organization</b>
Dick Bacon	University of Surrey
Niall Barr	Question Mark
Lance Blackstone	Pearson VUE
Jeanne Ferrante	ETS
Pierre Gorissen	SURF
Regina Hoag	ETS
Gopal Krishnan	Pearson VUE
Steve Lay	Cambridge Assessment
Rowin Young	CETIS



## Revision History

Version No.	Release Date	Comments
Base Document 2.1	14 October 2005	The first version of the QTI v2.1 specification.
Public Draft 2.1	9 January 2006	The Public Draft v2.1 of the QTI specification.
Public Draft 2.1 (revision 2)	8 June 2006	The Public Draft v2.1 (revision 2) of the QTI specification.

*IMS Global Learning Consortium, Inc. ("IMS/GLC") is publishing the information contained in this IMS Question and Test Interoperability Integration Guide ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.*

*IMS/GLC 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/GLC would appreciate receiving your comments and suggestions.*

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

*Please refer to Document Name: IMS Question and Test Interoperability Integration Guide Revision: 8 June 2006*

---