NLQTI: Dutch Profile for Question and Test Interoperability

Introduction and common sections

Version: V1.1 (August 2013)

This profile contains information derived from a subset of standards from IMS Global Learning Consortium (http://www.imsglobal.org). The standards, used with permission, include: IMS Content Packaging v1.1.3 (http://www.imsglobal.org/content/packaging/), IMS Question & Test Interoperability v2.1 (http://www.imsglobal.org/question/) and IMS Metadata v1.3 (http://www.imsglobal.org/metadata/).

Made possible by:
Kennisnet slo groep educatieve uitgeverijen Nederlands uitgeversverbond

Table of Contents

Colophon

Project team:	Jim Bijlstra; Jeroen Hamers; Marjolijn van Hooff; Erik Siegel
Author(s):	Erik Siegel; Wouter Huijnink
Consulted experts:	Wouter Huijnink; Bieke vd Korst
Consulted organizations:	Andriessen; Boom Test Uitgevers; Bureau ICE; Citaverde College; CITO; College van Examens; De Rode Planeet; Deviant; Edu'Actief; Efinity; Malmberg; Noordhoff Uitgevers; Orange11; Paragin; Platform VVVO;Roadside; SURF; ThiemeMeulenhoff; Threeships

Document history

Versie	Datum	Omschrijving
V0.9	7 december 2011	Eerste volledige en door de reviewgroep goedgekeurde versie, plateau 1
V0.91	April 2011	Uitwerking voor plateau 2
V0.92	Mei 2012	Kleine tekstuele bijstellingen n.a.v. expert groep bijeenkomst
V0.93	Sep 2012	Translated to English
V1.0	November 2012	Small textual changes and typos. Abbreviation changed to NLQTI. Colophon added. Final version.
V1.1	August 2013	NLQTI now based upon the QTI 2.1 Final version. A list of changes can be found in sect. 7/pg. 12.

1 Introduction

1.1 Background

Being able to easily exchange digital questions and tests is important within the Dutch educational system. It enables the exchange of content between educational institutes and suppliers of content management, learning and assessment software in an open and transparent way. Improving the exchange helps opening up formerly isolated sources of educational materials.
Digital questions and tests are made and used in a wide variety of systems and environments, using different data formats and processing models. Because of this there is a demand for a clear and easy to implement standard for exchanging the digital questions and tests. The international standard for this is “Question and Test Interoperability” (QTI) by IMS (current version: 2.1 Final; August 2012).
QTI however is complex and extensive, which makes implementation difficult, time-consuming and expensive. To overcome these disadvantages, Dutch educational institutions, together with partners from the publication and software industry, created a Dutch QTI profile: A much “lighter” version of the full QTI standard, geared towards the exchange of information and ease of implementation. Although created by and for the Dutch educational market, its applicability is probably much wider.

This document is part of the technical documentation set for the Dutch Profile for Question and Test Interoperability, a.k.a. NLQTI. The full set of documents is listed in sect. 2/pg. 4. Their goal is to supply content and software developers with enough information to implement this profile.
Main target groups are content management, learning and assessment software developers and people working with the (XML) content directly.
Familiarity with QTI and XML is assumed.

1.2 This document

This document, “NLQTI: Dutch profile for Question and Test Interoperability – Introduction and Common Parts”, contains general information, relevant to the rest of the profile:

- How to interpret the documents, notations used, and how to use the additional tools: sect. 3/pg. 5
- Some common/shared information: sect. 4/pg. 7
- Dealing with graphical design, styling and layout: sect. 5/pg. 8
- Dealing with assets: sect. 6/pg. 10

1.3 Points of departure

Creating a profile like this is a balancing act between leaving out enough to make it workable and keeping enough to keep it useable. We used the following points of departure as our guidelines:

- Focus on the needs of the Dutch primary, secondary and intermediate vocational education.
- Emphasis on exchangeability: This profile is primarily geared towards *exchanging* questions and tests between (very) different systems. Information about presenting and playing is secondary.
- Emphasis on ease of implementation: This profile aims to make implementation less complicated and time-consuming.
  Experience shows that some parts of QTI take a disproportional amount of time and effort to implement. Where this is the case this profile limits the options and replaces them with a fixed number of choices. An example is the “programming language” used in QTI for, among others, results processing. Within the profile, implementation of this language is not necessary. Another consequence is that support for the auditory and visual disabled is not (yet) present.
- Emphasis on testing (in contrast to exercising): Digital exercises are not widely used for those parts of the Dutch educational system this profile applies to. Exercising means support for adapting questions and tests based on how the learner progresses, which results in a huge complexity increase. Since no need for adaptive learning materials was (yet) perceived, it was left out of the profile.

2 Additional documents

2.1 Profile documents

The Dutch Profile for Question and Test Interoperability consists of the following documents:

[NLQTI-AB] NLQTI: Dutch Profile for Question and Test Interoperability - Algemene beschrijving toepassingsprofiel op basis van IMS QTI v2.1
Functional description of the profile, ***Dutch only****.*

[NLQTI-ICS] NLQTI: Dutch Profile for Question and Test Interoperability – Introduction and common sections
Common parts of the profile.

[NLQTI-ITEM] NLQTI: Dutch Profile for Question and Test Interoperability - Items
Description of items within this profile.

[NLQTI-TEST] Dutch Profile for Question and Test Interoperability - Tests
Description of tests within this profile

[NLQTI-CP] NLQTI: Dutch Profile for Question and Test Interoperability - Content Packaging
Describes the way items and tests should be packaged according to this profile.

2.2 Other documents and sources of information

The following documents/sources contain additional information. Some of these are in Dutch.

[CVE-MC] Mediastandaarden Computerexamensysteem
College voor Examens; Versie 1.0; 7 februari 2011

[IMSCP-BPG] IMS Content Packaging Best Practice Guide
Version 1.1.3 Final Specification; 2006

[IMSCP-INFO] IMS Content Packaging Information Model
Version 1.1.3 Final Specification; 2006

[IMSCP-XML] IMS Content Packaging XML Binding
Version 1.1.3 Final Specification; 2006

[IMSMD] IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata
Version 1.3 Final Specification; August 2006

[KN-PID] Concept afspraak Unieke Persistente Identifier voor Leermateriaal en Metadatarecord
Kennisnet; 6 juli 2011

[NL-LOM] NL-LOM (Dutch metadata profile)
9 juni 2011

[QTI-IIG] IMS Question and Test Interoperability Integration Guide
Version 2.1 Final; 31 August 2013

[QTI-IMPL] IMS Question and Test Interoperability Implementation Guide
Version 2.1 Final; 31 August 2013

[QTI-INFO] IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model
Version 2.1 Final; 31 August 2013

[QTI-IO] IMS Question and Test Interoperability Overview
Version 2.1 Final; 31 August 2013

[QTI-MDU] IMS Question and Test Interoperability Meta-data and Usage Data
Version 2.1 Final; 31 August 2013

[QTI-MIGR] IMS Question and Test Interoperability Migration Guide
Version 2.1 Final; 31 August 2013

[QTI-RR] IMS Question and Test Interoperability Results Reporting
Version 2.1 Final; 31 August 2013

[QTI-XML] IMS Question and Test Interoperability (QTI) XSD Binding
Version 2.1 Final; 31 August 2013

[WIKI-1] Semantic HTML
http://en.wikipedia.org/wiki/HTML#Semantic_HTML

[WIKI-2] CSS explained: http://en.wikipedia.org/wiki/CSS

[XML-SDT] XML Schema Part 2: Data types, Second Edition
(http://www.w3.org/TR/xmlschema-2/)

3 Interpretation, notation and tools

This chapter describes how to interpret the profile information and the notational conventions used. It also describes the tools belonging to the profile and how to use them.

3.1 Interpretation of the profile

All profiles and standards for the Dutch educational system are published and maintained by the Edustandaard organization. Most Edustandaard profiles (re-)describe standards in full, emphasizing the deviations from the original one.
QTI however is a very extensive standard. Completely re-describing it would have been far too much effort. Therefore this profile focusses on the deviations and limitations from the QTI standard. To aid the less QTI initiated, some limited functional descriptions were added.
Having said this, the following rules apply:

- When this profile does not limit a part of the QTI standard, the normal QTI rules apply.
- When this profile contradicts the QTI standard, the QTI rules apply. Of course this is considered an error and if you find one we would very much like to hear from you. Contact information can be found on the Edustandaard's website, www.edustandaard.nl.

3.2 Element and attribute descriptions

This profile is based on the QTI V2.1 XML binding. On top of that, additional rules and limitations for the XML elements and attributes apply. This is done using the following format:

Name	Prf?	M	Type	Remarks

- **Name**: The name of the element or attribute
- **Prf?** =Included in the profile? Is this element or attribute part of the profile? Values used here:
  - Yes = Part of the profile
  - Lim = (Limited) Element or attribute is part of the profile but limitations apply
  - No = Element or attribute is not part of the profile (and therefore should not occur)
- **M** = Multiplicity: The multiplicity of this element or attribute according to the profile (which might be more limited than QTI allows):
  - **1** = Single mandatory occurrence
  - **?** = 0 or 1 occurrences (single optional occurrence)
  - ***** = 0 or more occurrences (optional multiple occurrences)
  - **+** = 1 or more occurrences (multiple mandatory occurrences)
- **Type**: Data type according to the profile (which usually coincides with the QTI data type)
- **Remarks**: Any remarks about this element or attribute and its usage within the profile.

Most important for his profile is of course the **Prf?** column. What do the values in this column mean *exactly* for the profile's main target groups:

Prf?	Meaning for content producer:	Meaning for content consumer:
Yes	Is allowed to use (or for mandatory parts: must use) this element/attribute in the content produced.	Is obliged to use/interpret this element/attribute according to the QTI standard. Is not allowed to ignore it.
Lim	Is allowed to use (or for mandatory parts: must use) this element/attribute in the content produced, but must adhere to the limitations described.	Is obliged to use/interpret this element/attribute according to the QTI standard. Is not allowed to ignore it. When content is received that not adheres to the profile's limitations, the consumer must choose one of the following actions: - Interpret and use it according to the full QTI standard - Use a sensible value within the profile's limitations. Preferably a warning should be given.- Refuse the content (or the offending part of the content), providing an error message about this.

Prf?	Meaning for content producer:	Meaning for content consumer:
No	Must not use this element/attribute	When content is received that not adheres to the profile's limitations, the consumer should choose one of the following actions: - Interpret and use it according to the full QTI standard - Use a sensible value within the profile's limitations. Preferably a warning should be given. - Refuse the content (or the offending part of the content) providing an error message about this.

3.3 Tools

Accompanying these documents are various XML examples and tools (like for instance an adapted QTI XML Schema for this profile). You can find an overview of these in the `index.html` file in the profile's base directory.

3.3.1 Validation

The subdirectory `Validation` contains W3C XML Schemas and ISO Schematron files. To check XML files for full conformity to this profile, apply both.

Warning: When delivering XML content according to this profile that contain an explicit schema reference, do not refer to the specialised profile schema but to the general QTI schema (`imsqti_v2p1.xsd`) instead. This stops rendering engines that do not know the profile from rejecting the contents because they do not recognize the schema (name).

4 General subjects

4.1 Completeness of items and tests

- For this profile, QTI items and tests must be completely contained in a single file. It is not allowed to reference other XML files (with <xi:include>).
- There is one important exception to this rule: Referencing standard response processing templates from items. For more information: [NLQTI-ITEM], chap. 5.

4.2 MathML Support

QTI explicitly allows using MathML: the XML standard for mathematical formulas. MathML describes two variants: The presentation of a formula (how it should look) and the meaning of it (how to compute).
This profile only allows the *presentation* variant of MathML.

4.3 APIP Support

QTI supports the use of APIP (Accessible Portable Item protocol), but the NLQTI profile does not. Use of APIP extensions in NLQTI content is not allowed.

4.4 Tool identification and data

QTI supports adding tool specific data. For instance, some elements have an attribute called `label` for this. The maximum size for this data is limited to 256 characters.
The *presence* of this information is always optional. When present, the *interpretation* of the information must be optional as well: Rendering and processing the QTI item must always be possible without the use of this information.
This profile recommends to record at least the tool name and version the content was created with (`toolName` and `toolVersion` attribute). Rendering engines that recognize the values can adapt their processing based on this information. Other rendering engines should ignore it.

The above applies to the following elements/attributes:

- The attributes `toolName` and `toolVersion` on the root elements <assessmentItem> and <assessmentTest>. This should identify the content creation tool.
- The `label` attribute on various locations within the content. This attribute is meant to add rendering engine specific information.
- The elements `toolName`, `toolVersion` and `toolVendor` in the QTI specific metadata in Content Packages.

4.5 References

When a QTI XML file references other files (e.g. stylesheets, assets or items), the following rules apply. The underlying assumption is that the content, including assets, is contained in a Content Package (see [NLQTI-CP]) .

Notation	Example	Meaning
Relative path	`photo1.jpg` `img/photo1.jpg`	The referenced file is in the Content Package, relative to the referring file.
Absolute path without a protocol prefix	`/images/photo1.jpg`	The referenced file is in the Content package, relative to the root of the Content Package.
Other notation with protocol prefix	`http://…/photo1.jpg`	Assume the reference is a URL, referencing external content. See the remark about this below.

- Slashes (/) and backslashes (\) in file/path names are both allowed and should be interpreted identical (on all platforms)
- Relative paths are preferred
- References to external content (e.g. `http://…/foto.jpg`) are allowed but strongly discouraged. External references limit the usability of the content because an Internet connection must be present. Although common this is not always the case and there is no control over its quality.
  When an Internet connection is mandatory for playing the content, this should be recorded in the metadata (see [NLQTI-CP], chap. 4)

5 Graphical design and layout

Graphical design is a very tricky subject. Content producers would like to have full control over how content is presented to the user. However, this depends on the context and the environment and will not be the same in every player, on every device, etc. Because rendering engines can choose their own screen layouts, the enormous variety of devices (PC's, Laptops, Phones, Tablets, etc.) and the huge number of details involved, it is impossible to describe everything in full. Therefore this profile describes a number of guidelines and recommendations only.

5.1 Guidelines and recommendations

In general: It is recommended to stay away from graphical design and layout as far as is realistically possible. Use the available QTI and HTML tags for which they were designed for: Adding a semantic structure to the content that clarifies the intentions of it, not its layout.

QTI allows using stylesheets. Guidelines for stylesheets are:

- **Content creators:** Minimize the use of stylesheets. The probability of making things worse are larger than making things better.
- **Engine builders:** Perform active processing of the referenced stylesheets. Make sure this does not lead to erroneous targeting or hierarchy changes. These concepts are explained in sect. 5.2/pg. 8.

5.1.1 Guidelines QTI elements

Use QTI elements for structuring the content/information as much as possible. Using structural elements within an `<itemBody>` gives unambiguous information to the rendering engine.
For instance, separating background information and the actual question can be done by wrapping the background information in a `<rubricBlock>` element and the question in `<prompt>` element. Presenting this on A PC or Laptop is done in two columns. On a smartphone, some navigation is added that allows to use to easily switch between the two.

5.1.2 Guidelines XHTML elements

The (very) short summary: Use semantic HTML (more information: [WIKI-1]). This results in for instance:

- Use tables for which they were made for: presenting tabular information, not for forcing layout.
- Use header elements like `<h1>`, `<h2>`, etc. to add hierarchy to the content. Don't use arbitrary class names for the graphical effects only. For instance, `<span class="title_big">` in combination with a stylesheet could lead to the desired graphical effect. It does however not convey that this text is *semantically different* from the rest: that it is a header.
- Avoid elements with a more typographical meaning, use the corresponding semantic equivalents instead:
  - Not `<b>` but `<strong>`
  - Not `<i>` but `<em>`

5.2 Issues of using CSS in QTI

This paragraph contains a description of the most important issues using CSS (classes and stylesheets) within QTI.

In essence, QTI is a semantic standard: What the user sees is a conversion from semantic content to rendering in a certain environment. The final result depends on the environment, the device and the software used.
The QTI specification allows using stylesheets. This seems like it is indeed possible to control the final rendering: if you write your semantic content using `class` attributes, you should be able to control how the content will look like.
However, besides the fact that also CSS knows its restrictions (see [WIKI-2]), effective usage of style information is a lot harder to achieve than the specification suggests. The most important problems are:

- **Unknown context:** The biggest problem is that content creators are working blind when applying styles:
  - The size of the window the content is rendered in is unknown at design time. This could differ from 1024 pixels wide on a PC to 600 on a smartphone in landscape mode. And resolution is not all: A smartphone of 10cm wide with 960 pixels is completely different from a PC screen of 25cm wide with 1024 pixels. So defining absolute display positions and dimensions is not a good idea and will lead to poorer rendering (overlapping windows, a

large number of scrollbars, etc.) instead of the intended beautiful layout. The designer should use the principals of “liquid design”: Relative font sizes, relative positionings, floating windows, etc.

- It is absolutely not a given that QTI is rendered with an engine that supports CSS. It could for instance be a simple client program that does not use browser technologies.
- You have no control over which browser (engine), if any, will be used. So you’re not allowed to make assumptions about this designing your stylesheets. And creating good cross-browser stylesheets is hard.

- **Conversion results unknown:** Rendering engines have complete freedom converting the content for display. They will often adapt the content to its context.
  For instance: A rendering engine has reserved the use of <h1> and <h2> for itself (e.g. for subject and course name). So when rendering the QTI, it shifts the hierarchy of the headers down: <h1> becomes a <h3>, <h2> becomes a <h4>, etc. Now if you don’t apply the same shift to the stylesheets used, all the effort in creating a beautiful layout will have been wasted.
- **Targeting of elements:** Targeting within CSS is a difficult subject, both for content creators and rendering engine developers: CSS works with rules, which consists of selectors and some style declarations. A selector designates the content elements the style declarations should apply to. This mechanism is called targeting.
  As an example: if you add a rule for all <h1> elements, this rule will apply to all the <h1> elements presented. From the perspective of an engine builder, this is not acceptable: It could mean that the content creator could unintentionally force its complete house style on the application instead of on the question part only.
  To circumvent this, the content creator’s CSS definitions should target only elements within the content. You can achieve this in the following ways:
  - By using a specific class at the <itemBody> element and define all rules relative to the page element with this class. Whether or not this is going to work depends on whether the rendering engine will keep the <itemBody>’s class element intact.
  - By (auto-)converting the stylesheets delivered by the content creator, making all styling relative to a defined element/class

All this increases complexity a lot. Therefore: Stay away from stylesheets as much as possible.

6 Assets

Assets are defined as: All external objects referenced by QTI content, like images, films, sound, Word or PDF documents, etc. Within QTI this is done with the `<img>` or `<object>` element.

6.1 Functional limitations

- The rendering must do a best effort try to display the assets as intended (show images, run films, sound mp3's, etc.)
- When a rendering engine scales an asset (for instance because the display window area is limited), it should also provide the possibility to show the asset in full. For instance by opening a pop-up window after clicking on the image.
- Assets can be flagged to tell the rendering engine that the proportions/scaling of the asset are important for the viewer (sect. 6.3.2/pg. 11). If so, the rendering engine is not allowed to scale the asset or must make this very clear and provide a mechanism to display the asset in full.

6.2 Asset types

Giving the wide playing field of asset types and rendering engines, it is impossible to specify an exact and definite list of supported asset types. Therefore we provide recommendations only:

- Rendering engines should preferably be able to render the most widely used open and de-facto asset types:
  - For the Dutch market, a list of the most important open standards can be found in [CVE-MC]
  - It's advised to support PDF, GIF, MP3 and WAV type assets also
- Content creators are advised *not* to use:
  - Assets for which the type is closely coupled with the operating system (like WMA on Windows)
  - Assets for which the rendering depends on the installed client software (like Word or Excel)
  - Assets that are not supported on some of the more popular clients (like Flash which is not supported on iPad's)

Following these recommendations, rendering engines are advised to support at least the following asset types. Content producers should, to increase the interoperability of their content, limit themselves to this list.

- **Images:**
  - PNG (Portable Network Graphics)
  - JPEG (Joint Photographic Experts Group)
  - SVG (Scalable Vector Graphics)
  - GIF (Graphics Interchange Format)
  - PDF (Portable Document Format)
- **Video:**
  - Media container:
    - MKV (Matroska Multimedia Container)
    - MPEG-4 (Moving Pictures Experts Group)
  - Video codecs:
    - H264 / MPEG-4 AVC (Advanced Video Coding)
- **Audio:**
  - AAC / MP4 (Advanced Audio Coding)
  - MP3 (MPEG-1 of 2 Audio Layer III)
  - OGG (OGG Vorbis)
  - WAV

6.3 Technical implementation

6.3.1 QTI elements

*6.3.1.1 **<object>** element*

- Use the <object> element to reference an asset that is an *essential* part of the content.
- Attributes for <object>:

Name	Prf?	M	Type	Remarks
id	Yes	?	identifier	
class	Yes	?	string	See sect. 6.3.2/pg. 11
xml:lang	Yes	?	language	
label	Yes	?	string256	See sect. 4.4/pg. 7
xml:base	No			
data	Yes	1	uri	Reference to the asset. See sect. 4.5/pg. 7
type	Yes	1	string	MIME type of the asset
width	Yes	?	number or per-centage	The rendering engine is allowed to ignore these values
height	Yes	?		

- This profile also limits the contents of the <object> element. This is allowed to contain the following only:
  - <param> sub-elements for passing parameter values to the object
  - Other <object> elements. These should be treated as alternatives for the surrounding <object>. If the rendering engine is, for whatever reason, not able to show the surrounding <object> it should try the nested one. A nested <object> is of course allowed to contain a nested object itself, etc.
  - Text without markup. The rules for this are the same as for nested <object> elements: If the surrounding object cannot be shown, show the text.

*6.3.1.2 **<img>** element*

- Use the <img> element to reference an *optional* image, for instance a background image.
- Attributes on <img>:

Name	Prf?	M	Type	Remarks
id	Yes	?	identifier	
class	Yes	?	string	See sect. 6.3.2/pg. 11
xml:lang	Yes	?	language	
xml:base	No			
label	Yes	?	string256	See sect. 4.4/pg. 7
src	Yes	1	uri	Reference to the asset. See sect. 4.5/pg. 7
alt	Yes	1	string256	
longdesc	No			
height	Yes	?	fixed or percentage value	These values are hints. The rendering engine is allowed to ignore them
width	Yes	?		

6.3.2 Additional properties

- The class attribute is used to specify additional properties for an asset
- This profile defines the following values:

class value	Meaning
OBJECT_NOSCALE	The exact scaling of this object is meaningful for answering the question or test. Preferably it should be shown not scaled. If this is not possible it should be made clear to the learner that he/she is looking at a scaled image and some means should be given to access the not scaled version.

7 Appendix A: Changes between V1.0 and V1.1

In August 2012, after being in “Public Draft” status for years, IMS QTI finally became “Final”. The first official version of NLQTI (V1.0) was however still based on the QTI Public Draft version. Of course this had to change. We investigated the QTI Final version and found there were a large number of areas where the finalization of QTI influenced NLQTI. All these changes were gathered and this became the NLQTI V1.1.
This appendix provides an overview of the most important changes between NLQTI V1.0 and V1.1.

7.1 General changes

- All the names of all the schemas and Schematron files now end in `_v1p1` (instead of `_v1p0`) to reflect the version number change.
- The NLQTI V1.1 main schema file (`imsqti_v2p1_NLQTI_v1p1.xsd`) has a very different structure from the V1.0 one. This is because the QTI schema structure changed completely and all NLQTI changes were re-inserted into this new schema (not by hand, by a generator). However, this should not affect its usage.
- There is an extra schema `imsqti_metadata_v2p1_NLQTI_v1p1.xsd` for the metadata in Content Package manifest files. See sect. 7.4/pg. 12
- All examples are updated to reflect the V1.1 changes

7.2 Changes for Items

- An `<itemBody>` can now contain optional `<infoControl>` elements. These contain hints to the learner. Information that is not visible immediately but accessible to the learner by some action, for instance by clicking a button. This information can be viewed without a penalty on scoring. It can be used to supply hints.
- A `<mapEntry>` element can now be set to case-insensitive compare by the optional `caseSensitive` attribute.
- A `<rubricBlock>` can now contain `<stylesheet>` elements to provide stylesheet information for this rubric block only.
- The `adaptive` and `shuffle` attributes are now optional
- The original QTI response templates were changed to reflect the new situation. Their filenames are now in lowercase.

7.3 Changes for Tests

- An `<assessmentTest>` can now have `<stylesheet>` elements to a attach stylesheet information to a test.
- The `<timeLimits>` element now has an optional boolean `allowLateSubmission` attribute (default: `false`). Setting this to true means that the learners can still submit a result after the time limit has exceeded.
- The contents of the `<weight>` element is now empty.

7.4 Changes for Content Packaging

- The QTI metadata section for manifests is now in a separate namespace (`http://www.imsglobal.org/xsd/imsqti_metadata_v2p1`) and has a separate schema (`imsqti_metadata_v2p1_NLQTI_v1p1.xsd`).