Sharable Content Object Reference Model (SCORM®) 2004

SCORM Version 1.2 to SCORM 2004 Changes

Version 1.0

June 17, 2004

This page intentionally left blank.

Advanced Distributed Learning (ADL)
SCORM 2004
SCORM Version 1.2 to SCORM 2004 Changes

Version 1.0

Available at
www.adlnet.org

For questions and comments visit the
ADL Help & Info Center at ADLNet.org

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

Abstract

SCORM 2004 [1] contains many technical enhancements and changes from its predecessor, SCORM Version 1.2. This document provides a summary of those key changes. This document should not be treated as an exhaustive listing of all SCORM 2004 changes, but rather as a guide to be used with the SCORM 2004 documentation suite to understand the changes from SCORM Version 1.2 to SCORM 2004 and to determine what changes are needed to SCORM Version 1.2 products to migrate them from SCORM Version 1.2 to SCORM 2004.

This page intentionally left blank.

SECTION 1
SCORM Version 1.2 to SCORM 2004

This page intentionally left blank.

1.1. Introduction

This document offers a high-level overview of the key differences between SCORM Version 1.2 and SCORM 2004.

All of the SCORM books contain their own change history sections found in the appendices of the documents. As the individual books incorporate resolutions to defects and enhancements reported by the ADL Community, the change history in these books will be updated to reflect those additions.

Each book currently describes the changes based on the major specifications and standards that each book is based on. Refer to these specifications and standards for more information and specific requirements.

1.2. Summary of Changes

The key changes from SCORM Version 1.2 to SCORM 2004 are as follows:

- The addition of learning content sequencing capabilities as defined by the IMS Global Learning Consortium's Simple Sequencing (SS) specification to address the need for dynamic presentation of learning content based on learner performance.
- Updates to the IMS Content Packaging Specification.
- The approval of the IEEE's ECMAScript Application Programming Interface (API) and Learning Object Metadata (LOM) as formal IEEE standards and the addition of updates to these items to SCORM.
- Continuing standardization efforts on the IEEE Data Model for Content Object Communication and XML Schema Binding for LOM and the addition of updates to these items to SCORM.
- Various editorial and technical refinements based on feedback from the ADL Community on drafts and lessons learned in SCORM implementation.

More details about these changes can be found in this document and in the SCORM 2004 documentation suite.

1.3. Source of Changes

Since the release of SCORM Version 1.2, certain specifications and standards incorporated in SCORM have evolved. This section describes those underlying specifications and standards and how they have evolved from SCORM Version 1.2 to SCORM 2004.

1.3.1. IEEE Learning Technology Standards Committee (LTSC)

Since the release of SCORM Version 1.2, the IEEE LTSC Computer Managed Instruction (CMI) working group has been working to produce IEEE-accredited standards related to two aspects described in SCORM:

- Content to Learning Management System (LMS) communication
- Data model for content to LMS communication

The IEEE working group has produced the following two standards during this effort:

- IEEE 1484.11.1 – Draft Standard for Data model for Content Object Communication [2]
- IEEE 1484.11.2 – Standard for ECMAScript Application Program Interface (API) for Content to Runtime Service Communication [3]

These two standards capture the essence of the SCORM "Run-Time Environment" as described in the SCORM Run-Time Environment (RTE) book [1]. During SCORM Version 1.2 development, these two standards were just beginning the standardization effort within IEEE. At the time, SCORM Version 1.2 was based on a version of the CMI001 AICC/CMI Guidelines for Interoperability [4]. During the integration of the AICC specification into SCORM, ADL found discrepancies, issues and enhancements that were sent to AICC to be addressed. In time, AICC produced a newer version of the CMI Guidelines for Interoperability and submitted this work to IEEE to begin the standardization effort.

Since the release of SCORM Version 1.2, the IEEE LTSC Learning Object Metadata (LOM) working group has been developing IEEE accredited standards related to learning object meta-data. SCORM integrated this work as a means to describe certain components of the SCORM content model.

When SCORM Version 1.2 was published, the IEEE LTSC LOM working group was developing a standard for learning object meta-data. At the time, the IEEE 1484.12.1-Standard for Learning Object Meta-data [5] was in a draft form. ADL used the LOM XML Binding work that was going on inside the IMS Global Learning Consortium as a basis for SCORM Version 1.2 development (IMS Learning Resource Meta-data Specification [6]). IMS based the Extensible Markup Language (XML) Binding on a draft version of the IEEE LOM. Since the release of SCORM Version 1.2, IEEE has

approved the IEEE LOM as an accredited standard. During this time, IEEE continued to enhance and change the draft standard based on comments and issues brought forward during the standardization processes. The changes made to the standard are not reflected in the work performed by IMS.

Since the release of SCORM Version 1.2, IEEE also began a standardization effort on building an IEEE XML Binding of the LOM [7]. This effort was started before the release of SCORM 2004 and continues. The ongoing work continues to build the actual IEEE standard document and minor refinements to the actual XML Binding of the LOM.

1.3.2. IMS Global Learning Consortium Inc.

Since the release of SCORM Version 1.2, the IMS Global Learning Consortium, Inc. has updated some of the specifications that SCORM is built on. Specifically, the IMS Content Packaging Specification has gone through an IMS maintenance effort. This effort was done to update this specification set to fix defects encountered during product development and to clarify some ambiguities within the specification.

IMS has also produced a key specification referenced in SCORM 2004, namely the IMS Simple Sequencing Specification [8], which was in development for the last few years. This effort was taken up within IMS to address a critical need within the e-learning community for specifications to sequence discrete learning activities consistently to allow for the dynamic presentation and adaptability of learning content .

1.4. SCORM Content Aggregation Model (CAM)

1.4.1. SCORM Content Model Changes

Some clarifications were made to the SCORM Content Model to bring terminology defined in SCORM in line with terminology defined in the specifications SCORM is built upon. The term "content aggregation" was changed to "content organization" where content aggregation was meant to describe the organization of the content. A content aggregation is actually the aggregation of content (i.e., files, manifest, etc.) or the assembling, labeling and packaging of content. How the content is organized is not the aggregation, it is the "content organization." The CAM was updated throughout to ensure that the terms "content organization" and "content aggregation" were being used correctly.

Due to the clarifications that were made to "content aggregation" and "content organization," changes were also made to the names given to the different meta-data application profiles:

Table 1.4.1a: Meta-data Application Profile Changes

Manifest Element	SCORM 1.2 Meta-data Application Profile Name	SCORM CAM Version 1.3 Application Profile Name
<manifest>	Package-level Meta-data	Content Aggregation Meta-data
<organization>	Content Aggregation Meta-data	Content Organization Meta-data
<item>	Content Aggregation Meta-data	Activity Meta-data
<resource> where adlcp:scormType = "sco"	SCO Meta-data	No Change
<resource> where adlcp:scormType = "asset" <file>	Asset Meta-data	No Change

1.4.2. SCORM Content Packaging Changes

The changes to SCORM 2004 dealing with content package were based on several sources:

- Maintenance updates by IMS on the Content Packaging Specification.
- Changes required by the introduction of Sequencing and Navigation (SN).
- Changes required by the IEEE standardization effort on the SCORM Run-Time Environment Data Model.

Package Interchange File (PIF) Format: The IMS Content Packaging Specification was updated to reference an Internet Engineering Task Force (IETF) Request For Comment (RFC) dealing with the format of the PIF. The IMS specification was updated to reference RFC 1951 DEFLATE Compressed Data Format Specification [9].

Filename, Namespacing and Versioning: The IMS Content Packaging Specification and the XML Schema Definition (XSD) file were updated to align with IMS practices for schema development. The filename of the XSD was changed from `imscp_rootv1p1p2.xsd` to `imscp_v1p1.xsd`. Along with this change, IMS updated the namespace used to define the elements found in the XSD. The namespace changed from `http://www.imsproject.org/xsd/imcp_rootv1p1p2` to `http://www.imsglobal.org/xsd/imscp_v1p1`. The updates also affected the version number found in the XSD file. The version number, which is held by the `version` attribute of the `<xsd:schema>` element, was changed to `IMS CP 1.1.3` to reflect the version of the specification.

"xml:" Prefix Recommendation: The IMS Content Packaging Specification XSD (`imscp_v1p1.xsd`) was updated to change the reference to the xml:base declaration. The IMS Content Packaging Specification Version 1.1.2 XSD used the prefix x: to reference the xml:base attribute.

```
<xsd:attributeGroup name="attr.base">
   <xsd:attribute ref="x:base" />
</xsd:attributeGroup>
```

Code Illustration 1

```
<xsd:attributeGroup name="attr.base">
   <xsd:attribute ref="xml:base" />
</xsd:attributeGroup>
```

Code Illustration 2

This change was based on the recommendation of the World Wide Web Consortium (W3C).

Minimum and Maximum String Length Constraints:

The IMS Content Packaging Specification set and the IMS provided XSD files were not in sync with each other. The XSD was built to force maximum lengths for the following elements/attributes:

- `href` attribute of the `<resource>` and `<file>` elements was set at `2000` characters.
- `parameters` attribute of the `<item>` element was set at `1000` characters.
- `identifierref` attribute of the `<item>` element and `<dependency>` element was set at `2000` characters.
- `type` attribute of the `<resource>` element was set to `1000` characters.
- `structure` attribute of the `<organization>` element was set to `200` characters.
- `version` attribute of the `<manifest>` element was set to `20` characters.
- `<schema>` element was set to `100` characters.

- <schemaversion> element was set to 20 characters.
- <title> element was set to 200 characters.

If any of these elements or attributes contained values greater than their defined maximum lengths, then validating XML parsers would indicate that the XML values were invalid. These maximum lengths were not defined as maximum constraints in the IMS Content Packaging Specification set. The lengths were to be treated as a smallest permitted maximum (SPM), meaning that systems were responsible for supporting at least the number of characters. These systems could support more than the defined lengths or not (if they did not support more, the systems could do what they so desired with the characters past the length – e.g., truncate). The change was made to the XSD to remove these maximum lengths from the schema, which would permit values with more than the maximum lengths to pass XML validation.

Parameter Attribute Vocabulary: The IMS Content Packaging Specification was updated to add more information on the handling and processing of the parameters attribute found on an <item> element. The specification was updated to provide syntax for the parameters attribute that must be adhered to in manifests. An algorithm was also added for construction of the final href for the given resource (if the <item> referencing the <resource> had a parameters attribute defined).

IsVisible Attribute Clarification: The IMS Content Packaging Specification was updated to clarify the ambiguity on the application of the isvisible attribute for <item> elements. The specification was updated to indicate that the value only affects the item for which the isvisible attribute is defined and not the children of the <item> element (if the <item> element has children).

Type Attribute Vocabulary: The IMS Content Packaging Specification was updated to relax some of the constraints defined on the type attribute (attribute of the <resource> element). The following values are now defined in the IMS Content Packaging Specification as valid values for the type attribute:

- The term "webcontent," defined as content that can be hosted in or launched by a Web browser (this includes HTML-based content, content that requires plug-in and executables that are launched by a Web browser).
- The labels defined in Section 7 of the document "Using IMS Content Packaging to Package Instances of LIP and Other IMS Specifications, Version 1.0 Implementation Handbook, August 2001."
- The term "other" to be used when no other term is appropriate.

HREF File Name Format Recommendation: The IMS Content Packaging Specification was updated to reference an IETF RFC dealing with the format and syntax of the Href values found in a content package manifest. The IMS specification was updated to reference RFC 2396 [10].

(Sub)Manifest Usage Best Practice Clarification: The IMS Content Packaging Best Practice and Implementation Guide were updated to provide additional guidance and

requirements for the use of (sub)manifests. More clarification was added on scoping rules. These rules describe the scope of manifests and (sub)manifests explaining the rules of referencing elements across manifests. The specification was also updated to clarify what an `identifierref` was permitted to reference (with respect to the IMS Specification).

SCORM 2004 Content Packaging XML Binding Changes: The following table outlines the changes between a SCORM Version 1.2 Content Package Manifest and a SCORM 2004 Content Package Manifest.

Table 1.4.2a: Summary of Content Packaging Manifest Changes

SCORM Version 1.2	SCORM 2004
The namespace defined by IMS was: http://www.imsproject.org/xsd/imscp_rootv1p1p2	The IMS namespace was changed to: http://www.imsglobal.org/xsd/imscp_v1p1 This affects the namespace declarations found in an xml element.
The namespace define by ADL was: http://www.adlnet.org/xsd/adlcp_rootv1p2	The ADL namespace was changed to: http://www.adlnet.org/xsd/adlcp_v1p3 This affects the namespace declarations found in an xml element.
The <item> element that represents a parent (contains other <item> elements) could reference a resource found in the content package. The <item> element's identifierref attribute could contain a string that is equivalent to an identifier of a resource.	SCORM 2004 does not allow parent elements to reference resources, only leaf-items (<item> elements that do not contain other <item> elements) can reference resources.
The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:prerequisites>.	The <adlcp:prerequisites> element was removed from the ADL Content Packaging Extension XML Schema. The element no longer exists and should be replaced with appropriate sequencing rules.
The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:maxtimeallowed>.	The <adlcp:maxtimeallowed> element was removed from the ADL Content Packaging Extension XML Schema. The element no longer is permitted in SCORM 2004 Content Package manifests. The element should be replaced with the appropriate IMS sequencing rule (Refer to the SCORM Content Aggregtation Model book for more details).
The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:timelimitaction>.	The <adlcp:timelimitaction> element was removed from the ADL Content Packaging Extension XML Schema. The element was changed to <adlcp:timeLimitAction>. All SCORM 2004 Content Package Manifests should use the new element.

The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:datafromlms>.	The <adlcp:datafromlms> element was removed from the ADL Content Packaging Extension XML Schema. The element was changed to <adlcp:dataFromLMS>. All SCORM 2004 Content Package Manifests should use the new element.
The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:masteryscore>	The <adlcp:masteryscore> element was removed from the ADL Content Packaging Extension XML Schema. The element no longer is permitted in SCORM 2004 Content Package Manifests. The element should be replaced with the appropriate IMS sequencing rule (Refer to the SCORM Content Aggregation Model book for more details).
The <metadata> child element of the <manifest> element was optional.	Changed the requirement to require the <metadata> child element of the <manifest> element.
The <schema> child element of the <metadata> element was optional (i.e., the manifest.metadata.schema element).	Changed the requirement to require the <schema> element of the <metadata> element. The <schema> element is required to have a value of "ADL SCORM". The quotes are not part of the characterstring, they are used to delineate the value.
The <schemaversion> child element of the <metadata> element was optional (i.e., the manifest.metadata.schemaversion element).	Changed the requirement to require the <schemaversion> element of the <metadata> element. The <schemaversion> element is required to have a value of "CAM 1.3". The quotes are not part of the characterstring, they are used to delineate the value.
The <schema> and <schemaversion> elements could have been part of the <metadata> describing an organization (i.e., <organization> element).	The <schema> and <schemaversion> elements are not permitted to be used as child elements of a <metadata> element for the organization (i.e., <organization> element).
The <schema> and <schemaversion> elements could have been part of the <metadata> describing an item (i.e., <item> element).	The <schema> and <schemaversion> elements are not permitted to be used as child elements of a <metadata> element for the item (i.e., <item> element).
The <schema> and <schemaversion> elements could have been part of the <metadata> describing a resource (i.e., <resource> element).	The <schema> and <schemaversion> elements are not permitted to be used as child elements of a <metadata> element for the resource (i.e., <resource> element).
The <schema> and <schemaversion> elements could have been part of the <metadata> describing a file (i.e., <file> element).	The <schema> and <schemaversion> elements are not permitted to be used as child elements of a <metadata> element for the file (i.e., <file> element).
The ADL Content Packaging Extension XML Schema had an element declaration for <adlcp:scormtype>.	The <adlcp:scormtype> element was removed from the ADL Content Packaging Extension XML Schema. The element was changed to <adlcp:scormType>. All SCORM 2004 Content Package Manifest should use the new element.

1.4.3. SCORM Meta-data Changes

At the time of publication of SCORM Version 1.2, the IEEE P1484.12.1 Learning Object Metadata (LOM) was a draft standard. The IMS Learning Resource Meta-data was based on IEEE Working Draft 6.1 of the LOM. IMS provided the XML binding that was used in the SCORM Version 1.2. Before the release of SCORM 2004, the IEEE formally published the IEEE 1484.12.1-2002 LOM standard. The following table documents the changes from the IEEE Working Draft 6.1 of 1484.12.1 and the formal standard.

Table 1.4.3a: Summary of Meta-data XML Binding Changes

SCORM Version 1.2	SCORM 2004
The XML Binding namespace value was: http://www.imsglobal.org/xsd/imsmd_rootv1p2p1	The XML Binding namespace changed to: xmlns=http://ltsc.ieee.org/xsd/LOM
All elements that were composed of two or more words were bound to XML with lowercase names: • <aggregationlevel> • <lifecycle> • <datetime> • <metametadata> • <minimumversion> • <maximumversion> • <installationremarks> • <otherplatformrequirements> • <interactivitytype> • <learningresourcetype> • <interactivitylevel> • <semanticdensity> • <intendedenduserrole> • <typicalagerange> • <typicallearningtime> • <copyrightandotherrestrictions> • <taxonpath>	All elements that were composed of two or more words were bound to XML with lower camel case names: • <aggregationLevel> • <lifeCycle> • <dateTime> • <metaMetadata> • <minimumVersion> • <maximumVersion> • <installationRemarks> • <otherPlatformRequirements> • <interactivityType> • <learningResourceType> • <interactivityLevel> • <semanticDensity> • <intendedEndUserRole> • <typicalAgeRange> • <typicalLearningTime> • <copyrightAndOtherRestrictions> • <taxonPath>
The LangString data type was bound as: • <langstring xml:lang=”*langcode*”>	The LangString data type’s binding changed to: • <string language=”*langcode*”>

The Vocabulary Data Type was bound as: <source> <langstring xml:lang="x-none"></langstring> </source> <value> <langstring xml:lang="x-none"></langstring> </value>	The Vocabulary Data Type's binding changed to: <source></source> <value></value>
<general> Element	
The <identifier> element found as a child of the <general> element was reserved and not permitted to be used in a meta-data instance.	The <identifier> element is no longer reserved. The <identifier> element was also changed to include two subelements: • <catalog> • <entry> The <identifier> element has a smallest permitted maximum (SPM) of 10.
The <catalogentry> element existed. The <catalogentry> element contained two subelements: <catalog> and <entry>.	The <catalogentry> element has been deleted from the LOM information model and binding. The subelements of <catalogentry> moved to be subelements of <identifier>.
The <catalog> element existed as a subelement of <catalogentry> and was described as: "The name of the catalog (i.e. listing identification system)."	The <catalog> element moved to be subelement of <identifier> and the definition changed to: "The name or designator of the identification or cataloging scheme for this entry.
The <entry> element existed as a subelement of <catalogentry> and was described as: "Actual string value of the entry within the catalog (i.e. listing identification system)."	The <entry> element moved to be a subelement of <identifier> and the definition changed to "The value of the identifier within the identification or cataloging scheme that designates or identifies this learning object. A namespace specific string."
<lifecycle> Element	
The Entity element existed and was bound as: <centity> <vcard></vcard> </centity> No format was defined.	The <centity> element XML binding changed to: <entity></entity> vCard format was more formally specified.
The <identifier> element found as a child of the <lifeCycle> element was reserved and not permitted to be used in a meta-data instance.	The <identifier> element is no longer reserved. The <identifier> element was also changed to include two subelements: • <catalog> • <entry> The <identifier> element has a smallest permitted maximum (SPM) of 10.
The <catalogentry> element existed. The <catalogentry> element contained two subelements: <catalog> and <entry>.	The <catalogentry> element has been deleted from the LOM information model and binding. The subelements of <catalogentry> moved to be subelements of <identifier>.

The <catalog> element existed as a subelement of <catalogentry> and was described as: "The name of the catalog (i.e. listing identification system)."	The <catalog> element moved to be subelement of <identifier> and the definition changed to: "The name or designator of the identification or cataloging scheme for this entry.
The <entry> element existed as a subelement of <catalogentry> and was described as: "Actual string value of the entry within the catalog (i.e. listing identification system)."	The <entry> element moved to be a subelement of <identifier> and the definition changed to "The value of the identifier within the identification or cataloging scheme that designates or identifies this learning object. A namespace specific string."
<metaMetadata> Element	
The Entity element existed and was bound as: <centity> <vcard></vcard> </centity> No format was defined.	The <centity> element XML binding changed to: <entity></entity> vCard format was more formally specified.
The Metadata Schema element was bound as <metadataschema>. The element existed but no requirements on its use.	The Metadata Schema element's binding was changed to: <metadataSchema> Formal requirements on the use of the <metadataSchema> element. The element is required to exist 2 or More times. The value of this element shall be: • <metadataSchema>LOMv1.0</metadataSchema> • <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
<technical> Element	
The <orComposite> element did not exist.	The <orComposite> element was added as a subelement of <requirement>. The <orComposite> element has a SPM of 40. The <orComposite> now contains the following subelements (moved from the <requirements> element: • <type> • <name> • <minimumVersion> • <maximumVersion> The <orComposite> element is described as: Grouping of multiple requirements. The composite requirement is satisfied when one of the component requirements is satisfied, e.g., the logical connector is OR.
The <type> element was a subelement of <requirement>.	The <type> element was moved to be a subelement of <orComposite>.
The <name> element was a subelement of <requirement>.	The <name> element was moved to be a subelement of <orComposite>.

The <minimumVersion> element was a subelement of <requirement>.	The <minimumVersion> element was moved to be a subelement of <orComposite>.
The <maximumVersion> element was a subelement of <requirement>.	The <maximumVersion> element was moved to be a subelement of <orComposite>.
The <duration> element was bound as a DateTime Data Type: <duration> <datetime></datetime> <description> <langstring></langstring> </description> </duration>	The <duration> element was bound as a Duration Data Type: <duration> <duration></duration> <description> <string></string> </description> </duration>
<educational> Element	
The Typical Learning Time element <typicallearningtime> element was bound as a DateTime Data Type: <typicallearningtime> <datetime></datetime> <description> <langstring></langstring> </description> </typicallearningtime>	The Typical Learning Time element XML binding was changed to <typicalLearningTime> The <typicalLearningTime> element was bound as a Duration Data Type: <typicalLearningTime> <duration></duration> <description> <string></string> </description> </typicalLearningTime>
Meta-data Information Model Element: 5.9 TypicalLearningTime: DataType is Date	Meta-data Information Model Element: 5.9 TypicalLearningTime: DataType Change. Type is now Duration. Observe Duration change above.
<relation> Element	
The <identifier> element found as a child of the <relation> element was reserved and not permitted to be used in a meta-data instance.	The <identifier> element is no longer reserved. The <identifier> element was also changed to include two subelements: • <catalog> • <entry> The <identifier> element has a smallest permitted maximum (SPM) of 10.
The <catalogentry> element existed. The <catalogentry> element contained two subelements: <catalog> and <entry>.	The <catalogentry> element has been deleted from the LOM information model and binding. The subelements of <catalogentry> moved to be subelements of <identifier>.
The <catalog> element existed as a subelement of <catalogentry> and was described as: "The name of the catalog (i.e. listing identification system)."	The <catalog> element moved to be subelement of <identifier> and the definition changed to: "The name or designator of the identification or cataloging scheme for this entry.

The <entry> element existed as a subelement of <catalogentry> and was described as: "Actual string value of the entry within the catalog (i.e. listing identification system)."	The <entry> element moved to be a subelement of <identifier> and the definition changed to "The value of the identifier within the identification or cataloging scheme that designates or identifies this learning object. A namespace specific string."
<annotation> Element	
The Entity element existed and was bound as <person>. The Entity element's XML binding looked liked: <person> <vcard></vcard> </person> No format was defined.	The element's name changed to <entity> element XML binding changed to <entity>. <entity></entity> vCard format was more formally specified.
<classification> Element	
The <classification> element was required for Content Aggregation and SCO meta-data.	In SCORM 2004, the <classification> element was updated to be optional for all of the Meta-data Application Profiles.
The Taxon Path element was represented as <taxonpath> element.	The Taxon Path element is now represented as <taxonPath> element.
The <source> child element of the <taxonpath> element had a <langstring> child element: <taxonpath> <source> <langstring></langstring> </source> </taxonpath>	The <source> element was update to change the <langstring> element to a <string> element with a optional language attribute: <taxonpath> <source> <string language = "*language*"></string> </source> </taxonpath>
The <taxon> element was recursively nested: <taxonpath> <taxon> <taxon> <taxon></taxon> </taxon> </taxon> </taxonpath>	This recursive binding was changed to a flattened structure: <taxonpath> <taxon><taxon> <taxon></taxon> </taxon></taxon> </taxonpath>
The <entry> child element of the <taxon> element had a <langstring> child element: <taxon> <entry> <langstring></langstring> </entry> </taxon>	The <entry> element was update to change the <langstring> element to a <string> element with a optional language attribute: <taxon> <entry> <string language="*language*"></ string> </entry> </taxon>

ADL has provided an XML Stylesheet Language Translation (XSLT) to translate meta-data that was bound using IMS Learning Resource Metadata Version 1.2.1 to IEEE

1484.12.3 LOM XML Binding. This XSLT can be used to translate SCORM Version 1.2 meta-data to SCORM 2004 meta-data.

1.4.4. SCORM Sequencing and Navigation (SN)

The Sequencing and Navigation section of the CAM is new to SCORM. This section was added to the CAM to describe the conformance requirements for adding sequencing and navigation information to the content package manifest (imsmanifest.xml).

1.5. SCORM Run-Time Environment (RTE)

This section of the document describes the changes required to products based on the changes to the SCORM Run-Time Environment (RTE). As mentioned before the key source of these changes has been the standardization of the components of the API and SCORM Run-Time Environment Data Model. Since the release of SCORM Version 1.2, the Aviation Industry CBT Committee (AICC) submitted their specification set to the IEEE to begin the standardization process. During this time, the IEEE LTSC decided to begin the standardization process on two aspects of the submission from AICC:

- JavaScript Application Programming Interface
- Data Model used for Content Object Communication

The two standards that have been created since then are as follows:

- IEEE P1484.11.1 – Draft Standard for Data model for Content Object Communication
- IEEE 1484.11.2 – Standard for ECMAScript API for Content to Runtime Service Communication

This section describes the changes to SCORM 2004 based on this standardization process. To balance the need to support existing implementations with the need to make technical corrections and support emerging practice, the IEEE standard:

- Selectively includes those data elements from the CMI specification that are commonly implemented;
- Renames some data elements taken from the CMI specification to clarify their intended meaning;
- Modifies the data types of data elements taken from the CMI specification to reflect ISO standard data types and internationalization requirements;
- Removes some organizational structures used in the CMI specification to group data elements that are specific to the AICC community of practice and not generally applicable; and
- Introduces some data elements not present in the CMI specification to correct known technical defects in data elements taken from that specification.

1.5.1. General API Changes

The following list outlines some general changes to the API.

- An API Implementation's object name changed from API to API_1484_11.
- A new version attribute shall be provided for the API Implementation object. The attribute value shall consist of a period-delimited string containing major and

minor release values. The major version number shall be “1” and the minor version number shall be “0”. The first three characters, therefore, shall be “1.0”.

1.5.2. SCORM API Method Changes

The following list outlines the changes to the API Implementation method calls. This set affects both LMS implementations and SCO development. SCORM 2004 LMS implementations shall support these changes in their SCORM 2004 API Implementations. SCORM 2004 SCOs shall make these changes to their method calls.

- LMSInitialize(“”) changed to Initialize(“”)
- LMSFinish(“”) changed to Terminate(“”)
- LMSGetValue(*parameter*) changed to GetValue(*parameter*)
- LMSSetValue(*parameter_1,parameter_2*) changed to SetValue(*parameter_1,parameter_2*)
- LMSCommit(“”) changed to Commit(“”)
- LMSGetLastError() changed to GetLastError()
- LMSGetErrorString(*parameter*) changed to GetErrorString(*parameter*)
- LMSGetDiagnostic(*parameter*) changed to GetDiagnostic(*parameter*)

1.5.2.1 API Functional Behavioral Changes

Error code determination

SCORM 2004 now defines more specific error codes (refer to Section 1.4.3 Error Code Changes) for more situations than what was defined in SCORM Version 1.2. These error codes and the situations shall be implemented by the LMS.

Several error conditions have been added to SCORM 2004 that were not defined in SCORM Version 1.2. These error conditions are different from the set of error codes in SCORM Version 1.2. In some cases, a generic error code (301 – General Get Failure or 351 – General Set Failure) is set. In these situations, SCORM recommends a behavior for an LMS if a GetDiagnostic() method call is made by an LMS. These are only recommendations and the LMS may elect to follow them.

Initialize(“”)

In SCORM 2004, SCOs are permitted to call Initialize(“”) more than once. There is now a defined behavior for an LMS on how to process the duplicate calls.

GetErrorString()

In SCORM 2004, LMSs are no longer required to return a specific characterstring related to the current error code. The LMS can return any characterstring representation of descriptive information related to the error code. The error code determined is the important requirement. LMSs can relate any textual description the specific error code.

1.5.3. SCORM Error Code Changes

The following table maps the changes to the error codes that can be generated during the processing of various API method calls.

Table 1.5.3a: Summary of Error Code Changes

SCORM 1.2 Error Code	SCORM 2004 Error Code
0 – No error	0 – No error
101 – General Exception	101 – General Exception
	102 – General Initialization Failure
	103 – Already Initialized
	104 – Content Instance Terminated
	111 – General Termination Failure
	112 – Termination Before Initialization
	113 – Termination After Termination
	122 – Retrieve Data Before Initialization
	123 – Retrieve Data After Termination
	132 – Store Data Before Initialization
	133 – Store Data After Termination
	142 – Commit Before Initialization
	143 – Commit After Termination
201 - Invalid argument error	201 – General Argument Error
202 - Element cannot have children 203 - Element not an array. Cannot have count	301 – General Get Failure
	351 – General Set Failure
	391 – General Commit Failure
401 - Not implemented error	401 – Undefined Data Model Element
401 - Not implemented error	402 – Unimplemented Data Model Element
301 - Not initialized	403 – Data Model Element Value Not Initialized
403 - Element is read only	404 – Data Model Element Is Read Only
404 - Element is write only 402 - Invalid set value, element is a keyword	405 – Data Model Element Is Write Only
405 - Incorrect Data Type	406 – Data Model Element Type Mismatch
	407 – Data Model Element Value Out Of Range

	408 – Data Model Dependency Not Established

1.5.4. SCORM Run-Time Environment Data Model Changes

1.5.4.1 General Changes and Enhancements

The following is a listing of general changes and enhancements to the SCORM Run-Time Environment Data Model:

1. Removed the concept of the "core" data model category. All elements that were bound as cmi.core.<some_element> were flattened to cmi.<some_element>.
2. Removed the concept of the "student_data" data model category. All elements that were bound as cmi.student_data.<some_element> were flattened to cmi.<some_element>.
3. All AICC-defined data types were changed to ISO 11404 [11] Data Types.
4. Requirements and guidance were added to data model elements that impact sequencing decisions made by an LMS.
5. More specific requirements and guidance were provided in handling elements that represent collection of data (Comments From Leaner, Comments From LMS, Objectives and Interactions).
6. Removed hard maximum characterstring lengths. These characterstring lengths were changed to SPMs.
7. Added SPMs for data model elements representing collections. There is now a minimum requirement support for data model elements representing these collections.
8. Introduction of Reserved Delimiters to represent certain data model characteristics found in the IEEE Data Model that are needed for the SCORM "dot-notation" binding.
9. Explicit requirement changes to support international charactersets for those data model elements that need localization information.
10. Characterstring are now required to be implemented to support ISO 10646 [12].
11. Dates and Time intervals now have a specific format required when communicating dates and time intervals with an LMS.
12. Explicit requirements in defining extensions to the data model and requirements on processing extensions were added.
13. Definition clean up to match definitions defined in IEEE Data Model.
14. All Run-Time Environment Data Model elements are now required to be implemented (mandatory) by LMSs.

1.5.4.2 Run-Time Data Model Elements

At the time of publication of SCORM Version 1.2, the SCORM Run-Time Environment Data Model was based on the AICC CMI001 Guidelines for Interoperability Version 3.4. During the development of SCORM 2004, the AICC submitted their guidelines to IEEE for standardization. The changes to the SCORM Run-Time Environment Data Model were due to this standardization process, lessons learned, bugs found with SCORM

Version 1.2 and ADL Community requests for change. The following table documents the changes from the SCORM Version 1.2 to SCORM 2004.

Table 1.5.4.2a: Summary of RTE Data Model Changes

SCORM Version 1.2 Data Model Element	Summary of Changes in SCORM 2004
cmi.core._children	Element removed. The concept of a core data model category was removed.
cmi.core.student_id	Changed the SCORM dot-notation binding to cmi.learner_id Changed the data type from CMIIdentifier to long_identifier_type (see the SCORM Run-Time Environment book for more information on the long_identifier_type) The format of the value held by the data model element is required to be a valid Uniform Resource Identifier (URI) [10] and recommended to be a Uniform Resource Name (URN) [13]. A smallest permitted maximum has been defined for the length of the value. The SPM is defined to be 4000 characters. Error code changed for cases where a SCO tries to set cmi.learner_id (From 403 - 404) • From: SCORM Version 1.2 – 403 Element is read only • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.student_name	Changed the SCORM dot-notation binding to cmi.learner_name Data type changed from CMIString255 to localized_string_type (see SCORM Run-Time Environment for more information on the localized_string_type). Removed the requirement of being 255 characters or less. The data model element is now defined to have an SPM of 250 characters. Restriction on format (last name, first name middle initial) was removed. Implementations may provide any format desired. Error code changed for cases where a SCO tries to set cmi.learner_name (From 403 - 404)

	• From: SCORM Version 1.2 – 403 Element is read only • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.lesson_location	Changed the SCORM dot-notation binding to cmi.location Changed the data type from CMIString255 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring) Removed the requirement of being 255 characters or less. The data model element is now defined to have an SPM of 1000 characters. Value space for the characterstring has to be valid characters as defined in ISO10646-1:2000 [12]. Error code addition for cases where an SCO calls GetValue() before the value has been set by the SCO. The LMS is responsible for returning an empty characterstring and setting the error code to 403 – Data Model Element Value Not Initialized. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.credit	Changed the SCORM dot-notation binding to cmi.credit Changed the data type declaration from CMIVocabulary to state (credit, no_credit). This does not affect that actual binding of the state to its respective characterstring representations ("credit", "no-credit" are still the restricted vocabulary tokens). Note that the quotes are not part of the restricted token, they are used to delineate the actual values of credit. Error code changed for cases where a SCO tries to set cmi.credit (From 403 - 404) • From: SCORM Version 1.2 – 403 Element is read only • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Added more information on the relationship between cmi.credit and cmi.mode.
cmi.core.lesson_status	This data model element still exists in the IEEE 1484.11.1

	Data Model for Content Object Communication (defined as Lesson Status). The IEEE standard suggests not using this element because of potential to remove from a future edition of the standard. The element has historically caused problems because it holds dual meanings (completion and mastery). IEEE suggests using the two new elements: • success_status • completion_status SCORM 2004 does not require the support of cmi.core.lesson_status in a SCORM 2004 LMS. However, LMS may need to support the value to support legacy content. SCORM 2004 SCOs should not use Lesson Status.
cmi.success_status	This is a new element for SCORM 2004. It was introduced in IEEE to remove the ambiguity found in the SCORM Version 1.2 data model element cmi.core.lesson_status. This data model element replaces cmi.core.lesson_status (i.e., the mastery status portion). This data model element supports the cmi.core.lesson_status vocabulary values of "passed", "failed". Also supports a new vocabulary of "unknown". Note the quotes are not part of the restricted token, they are used to delineate the actual value of the success status. Added more guidance and requirements for processing and determining cmi.success_status. Specifically, rules that permit the LMS to override any value stored by the SCO. The default value of “unknown” is defined. This value shall be assumed until either a SCO sets this value or some processing by an LMS overrides this value. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.completion_status	This is a new element for SCORM 2004. It was introduced in IEEE to remove the ambiguity found in the SCORM Version 1.2 data model element cmi.core.lesson_status. This data model element replaces cmi.core.lesson_status (i.e., the completion status portion). This data model element supports the cmi.core.lesson_status vocabulary values of "completed", "incomplete" and “not_attempted”. Also supports a new vocabulary of "unknown". Note the quotes are not part of the restricted token, they are used to delineate the actual value of the completion status. The data model element does not contain the restricted vocabulary token of "browsed" any more. This vocabulary value was dropped from the list of valid values within IEEE.

	Added more guidance and requirements for processing and determining cmi.completion_status. Specifically rules that permit the LMS to override any value stored by the SCO. The default value of “unknown” is defined. This value shall be assumed until either a SCO sets this value or some processing by an LMS overrides this value. Details on how this element impacts sequencing was added. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.entry	Changed the SCORM dot-notation binding to cmi.entry. Changed the data type declaration from CMIVocabulary to state (ab_initio, resume, _nil_). This does not affect that actual binding of the state to its respective characterstring representations ("ab-initio", "resume" and “”, empty characterstring, are still the restricted vocabulary tokens). Note that the quotes are not part of the restricted token, they are used to delineate the actual value of the entry data model element. More guidance and requirements added to describe the LMS’s responsibilities for initializing and determining this value. Error code changed for cases where a SCO tries to set cmi.entry (From 403 - 404) • From: SCORM Version 1.2 – 403 Element is read only • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.score._children	Changed the SCORM dot-notation binding to cmi.score._children. Changed the data type from CMIString255 to characterstring (see SCORM Run-Time Environment for more information on the characterstring) Removed the requirement of being 255 characters or less. Updated requirement to indicate that there is no requirement on the order of the elements in the comma-separated list. Updated list of children elements to include new data model element scaled. Error code changed for cases where a SCO tries to set cmi.score._children (From 402 - 404)

	• From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.score.raw	Changed the SCORM dot-notation binding to cmi.score.raw. Changed the data type from CMIDecimal or CMIBlank to real (10,7) (see SCORM Run-Time Environment for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the raw score being a value between 0 and 100. Removed the following statements found in the Usage section: o "When the student is in the first attempt at a SCO, the cmi.core.score.raw should be set to "" empty string"). o "If no cmi.core.score.raw was set in a SCO and a request for the cmi.core.score.raw was made by the SCO, then an empty string should be returned". Error code changed for cases where a SCO tries to set cmi.score.raw and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Error code changed for cases where a SCO tries to set cmi.score.raw with a value that is not of the correct data type (From 405 - 406) Added error code for cases where SCO tries to retrieve the raw score prior to it being set (403 – Data Model Element Value Not Initialized).
cmi.core.score.min	Changed the SCORM dot-notation binding to cmi.score.min. Changed the data type from CMIDecimal or CMIBlank to

	real (10,7) (see the SCORM Run-Time Environment book for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the minimum score being a value between 0 and 100. Updated the behavior for cases where the SCO tries to retrieve the cmi.score.min prior to the value being set. For cases where a SCO requested the minimum score prior to the minimum score being set, the LMS should set an error code (403 – Data Model Element Value Not Initialized) and return an empty characterstring. The SCO can determine by the error code that the value has not been set yet. Error code changed for cases where a SCO tries to set cmi.score.min and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Added error code for cases where SCO tries to retrieve the minimum score prior to it being set (403 – Data Model Element Value Not Initialized).
cmi.core.score.max	Changed the SCORM dot-notation binding to cmi.score.max. Changed the data type from CMIDecimal or CMIBlank to real (10,7) (see the SCORM Run-Time Environment book for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the maximum score being a value between 0 and 100. Updated the behavior for cases where the SCO tries to retrieve the cmi.score.max prior to the value being set. For cases where a SCO requested the maximum score prior to the maximum score being set, the LMS should set an error code (403 – Data Model Element Value Not Initialized) and return an empty characterstring. The SCO can determine by the error code that the value has not been set yet. Error code changed for cases where a SCO tries to set cmi.score.max and the value is not a valid real number (From 405 - 406)

	• From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Added error code for cases where SCO tries to retrieve the maximum score prior to it being set (403 – Data Model Element Value Not Initialized).
cmi.score.scaled	New element added in SCORM 2004. See the SCORM Run-Time Environment (RTE) book for details about this element.
cmi.core.total_time	Changed the SCORM dot-notation binding to cmi.total_time. Changed the data type from CMITimespan to timeinterval (second,10,2) (see SCORM Run-Time Environment for more information on the timeinterval (second,10,2)) Revised language dealing with the temporal model (refer to *Section 2.1.1: Run-Time Environment Temporal Model* of the SCORM Run-Time Environment book for more details on the temporal model). This revision has also defined the requirements for calculating cmi.total_time more clearly. Changed the format to reflect the recommended format found in the IEEE standard. o From: HHHH:MM:SS.SS o To: P[yY][mM][dD][T[hH][mM][s[.s]S]] The initialization value for this element has changed, due to the type change, from 0000:00:00.00 to PT0H0M0S. Error code changed for cases where a SCO tries to set cmi.total_time (From 403 - 404) • From: SCORM Version 1.2 – 403 Element is read only. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.core.lesson_mode	Changed the SCORM dot-notation binding to cmi.mode. Changed the data type declaration from CMIVocabulary to state (browse, normal, review). Refer to SCORM Run-Time Environment for more information on the state data type. This does not affect that actual binding of the state to

	its respective characterstring representations ("browse", "normal" and “review” are still the restricted vocabulary tokens). Note that the quotes are not part of the restricted token, they are used to delineate the actual value of the entry data model element. Error code changed for cases where a SCO tries to set cmi.mode (From 403 - 404) • From: SCORM Version 1.2 – 403 Element is read only • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Details on how this element impacts sequencing was added. Added more information on the relationship between cmi.credit and cmi.mode.
cmi.core.exit	Changed the SCORM dot-notation binding to cmi.exit. Changed the data type declaration from CMIVocabulary to state (timeout, suspend, logout, normal, _nil_). Refer to the SCORM Run-Time Environment for more information on the state data type. This does not affect that actual binding of the state to its respective characterstring representations ("time-out", "suspend", “logout”, “normal” and “”, empty characterstring,. Note that the quotes are not part of the restricted token, they are used to delineate the actual value of the entry data model element. Added the additional vocabulary token “normal” to describe a normal exit. In SCORM Version 1.2 the empty characterstring was used to represent two types of exits: normal and undetermined. SCORM 2004 split this semantics up into two distinct values “normal” and “”. Clarified the meaning of "logout". It does not mean logging the learners out of the LMS and making them reauthenticate. Added more specific behavior logic (in relation to cmi.entry). Details on how this element impacts sequencing was added. Error code changed for cases where a SCO tries to retrieve cmi.exit (From 404 - 405) • From: SCORM Version 1.2 – 404 Element is write only • To: SCORM 2004 - 405 Data Model Element Is Write Only

	Error code changed for cases where a SCO tries to set cmi.mode and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch
cmi.core.session_time	Changed the SCORM dot-notation binding to cmi.session_time. Changed the data type from CMITimespan to timeinterval (second,10,2) (see SCORM Run-Time Environment for more information on the timeinterval (second, 10, 2)) Revised language dealing with the temporal model (refer to *Section 2.1.1: Run-Time Environment Temporal Model* of the SCORM RTE book for more details on the temporal model). This revision also defined the requirements for using cmi.session_time and its impacts on the cmi.total_time and the temporal model more clearly. Changed the format to reflect the recommended format found in the IEEE standard. o From: HHHH:MM:SS.SS o To: P[yY][mM][dD][T[hH][mM][s[.s]S]] Error code changed for cases where a SCO tries to getcmi.session_time (From 404- 405) • From: SCORM Version 1.2 – 404 Element is write only. • To: SCORM 2004 - 405 Data Model Element Is Write Only. Error code changed for cases where a SCO tries to set cmi.session_time and the value is not a valid time interval (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.suspend_data	Changed the data type from CMIString4096 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring). The restriction on the size of the characterstring was relaxed. The value is no longer a maximum length of 4096 (as defined by SCORM Version 1.2). The length is now defined as an SPM of 4000 characters.

	The LMS is no longer required to initialize this data model element to an empty characterstring. The behavior is now defined. If a SCO requests to retrieve cmi.suspend_data from the LMS and the data has not been set yet, the LMS is required to return an empty characterstring and set the error code to 403 – Data Model Element Value Not Initialized. More clarification, guidance and requirements were added to the handling and processing of cmi.suspend_data. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.launch_data	Changed the data type from CMIString4096 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring). The restriction on the size of the characterstring was relaxed. The value is no longer a maximum length of 4096 (as defined by SCORM Version 1.2). The length is now defined as an SPM of 4000 characters. If a SCO requests to retrieve cmi.launch_data from the LMS and the data has not been set, the LMS is required to return an empty characterstring and set the error code to 403 – Data Model Element Value Not Initialized. More clarification, guidance and requirements were added to the handling and processing of cmi.launch_data. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.comments	The cmi.comments data model element was updated to cmi.comments_from_learner. The cmi.comments_from_learner now contains several sub-elements: • comment • location • date_time The sub-elements were created to single out the pieces of a comment that were being used in legacy content. This data model element shall be supported by an LMS. The data model element is now represented as a collection of comments (array notation – as used by the SCORM dot.notation binding). In SCORM Version 1.2, if multiple comments were set by a SCO, the LMS was responsible for concatenating the comments. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.

	Refer to the SCORM Run-Time Environment book for more details on this data model element.
cmi.comments_from_lms	The cmi.comments_from_lms now contains several sub-elements: • comment • location • date_time The sub-elements were created to single out the pieces of a comment. This data model element shall be supported by an LMS. The data model element is now represented as a collection of comments (array notation – as used by the SCORM dot.notation binding). Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS. Refer to the SCORM Run-Time Environment book for more details on this data model element.
cmi.objectives._children	Changed the data type from CMIString255 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring) Removed the requirement of being 255 characters or less. Updated the requirement to indicate that there is no requirement on the order of the elements in the comma-separated list. Updated list of children elements to include new data model element success_status, completion_status and description. Error code changed for cases where a SCO tries to set cmi.objectives._children (From 402 - 404) • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives._count	Changed the data type from CMIInteger to non-negative integer (see the SCORM Run-Time Environment book for more information on the non-negative integer) Error code changed for cases where a SCO tries to set cmi.objectives._count • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword.

	• To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.id	Changed the data type from CMIIdentifier to long_identifier_type (see the SCORM Run-Time Environment book for more information on the long_identifier_type) Added requirements, based on new type, for the identifier to be a Universal Resource Identifier (URI) and recommendation of the value being a Universal Resource Name (URN). A smallest permitted maximum has been defined for the length of the value. The SPM is defined to be 4000 characters. Added additional requirements on how the cmi.objectives.n.id data model element shall be initialized using the <imsss:objectives> element found in an imsmanifest.xml file. The SCO developer needs to ensure that the identifiers are uniquely identifiable within at least the scope of the SCO. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.objectives.n.id and the value is not a valid long_identifier_type, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2, this error code was 405 – Incorrect data type. Added guidance on the usage of identifiers for the objectives element. The identifiers are what uniquely

	identifies one objective from another and the identifier should be used to guarantee that the correct set of data is being updated (not the position in the array). Added a requirement that the cmi.objectives.n.id be the first element set out of the set of objective data. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.score._children	Changed the data type from CMIString255 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring) Removed the requirement of being 255 characters or less. Updated requirement to indicate that there is no requirement on the order of the elements in the comma-separated list. Updated list of children elements to include new data model element scaled. Error code changed for cases where a SCO tries to set cmi.objectives.n.score._children (From 402 - 404) • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.score.raw	Changed the data type from CMIDecimal or CMIBlank to real (10,7) (see the SCORM Run-Time Environment book for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the objectives raw score being a value between 0 and 100. If the SCO invokes a GetValue() request to retrieve the objectives raw score prior to the raw score being set, then the LMS shall set the error code (403 – Data Model Element Value Not Initialized) and return an empty characterstring. Error code changed for cases where a SCO tries to set cmi.objectives.n.score.raw and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element

	Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.score.min	Changed the data type from CMIDecimal or CMIBlank to real (10,7) (see the SCORM Run-Time Environment book for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the objectives minimum score being a value between 0 and 100. If the SCO invokes a GetValue() request to retrieve the objectives minimum score prior to the minimum score being set, then the LMS shall set the error code (403 – Data Model Element Value Not Initialized) and return an empty characterstring. Error code changed for cases where a SCO tries to set cmi.objectives.n.score.min and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.score.max	Changed the data type from CMIDecimal or CMIBlank to real (10,7) (see the SCORM Run-Time Environment book for more information on the real (10,7)) Removed the requirement that an LMS initialize this value to an empty characterstring (“”). Removed the requirement of the objectives maximum score being a value between 0 and 100. If the SCO invokes a GetValue() request to retrieve the objectives maximum score prior to the maximum score being set, then the LMS shall set the error code (403 – Data Model Element Value Not Initialized) and return an empty characterstring. Error code changed for cases where a SCO tries to set cmi.objectives.n.score.max and the value is not a valid real number (From 405 - 406) • From: SCORM Version 1.2 – 405 Incorrect Data Type • To: SCORM 2004 - 406 Data Model Element Type Mismatch

	Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.score.scaled	New element added in SCORM 2004. See the SCORM Run-Time Environment book for more details on this element.
cmi.objectives.n.status	This data model element still exists in the IEEE 1484.11.1 Data Model for Content Object Communication (defined as Lesson Status). The IEEE standard suggests not using this element because of potential to remove from a future edition of the standard. The element has historically caused a lot of problems due to the fact that it holds dual meaning (completion and mastery). IEEE suggests using the two new elements: • success_status • completion_status SCORM 2004 does not require the support of cmi.objectives.n.status in a SCORM 2004 LMS. However, LMS may need to support the value to support legacy content. SCORM 2004 SCOs shall not use Status element.
cmi.objectives.n.success_status	This is a new element for SCORM 2004. It was introduced in IEEE to remove the ambiguity found in the SCORM Version 1.2 data model element cmi.objectives.n. status. This data model element replaces cmi.objectives.n.. status (i.e., the mastery status portion). This data model element supports the cmi.objectives.n.status vocabulary values of "passed", "failed". Also supports a new vocabulary of "unknown". Note the quotes are not part of the restricted token, they are used to delineate the actual value of the success status. The default value of "unknown" is defined. This value shall be assumed until either a SCO sets this value. Details on how this element impacts sequencing was added. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.completion_status	This is a new element for SCORM 2004. It was introduced in IEEE to remove the ambiguity found in the SCORM Version 1.2 data model element cmi.objectives.n.status. This data model element replaces cmi.objective.n.status (i.e., the completion status portion). This data model element supports the cmi.objectives.n.status vocabulary values of "completed", "incomplete" and "not_attempted". Also supports a new

	vocabulary of "unknown". Note that the quotes are not part of the restricted token, they are used to delineate the actual value of the completion status. The data model element does not contain the restricted vocabulary token of "browsed" any more. This vocabulary value was dropped from the list of valid values within IEEE. Added more guidance and requirements for processing and determining cmi.completion_status. Specifically rules that permit the LMS to override any value stored by the SCO. The default value of "unknown" is defined. This value shall be assumed until either a SCO sets this value or some processing by an LMS overrides this value. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.objectives.n.description	This is a new element defined for SCORM 2004. Information on the specifics of this data model element can be found in the SCORM Run-Time Environment book.
cmi.student_data	
cmi.student_data._children	Element removed. The concept of a student_data data model category was removed.
cmi.student_data.mastery_score	Changed the SCORM dot-notation binding to cmi.scaled_passing_score. Changed the data type from CMIDecimal to real (10,7) (see the SCORM Run-Time Environment book for more information on the long_identifier_type). The value of the element shall be in a range of -1 to 1. The data model element initialization procedure was updated to either: • The value stored in the <imsss:minNormalizedMeasure> element associated with the <imsss:primaryObjective> element for the <imscp:item> element that references a SCO resource, or • if the <imsss:minNormalizedMeasure> element has not been defined., the LMS shall not assume any initialization value. LMS behavior was added for cases where the SCO tries to get the cmi.scaled_passing_score and there was no <imsss:minNormalizedMeasure> defined. The LMS is responsible for setting the error code to 403 – Data Model Element Not Initialized and returning an empty characterstring. Error code changed for cases where a SCO tries to set cmi.scaled_mastery_score and the value is not a valid vocabulary token (From 405 - 406)

	• From: SCORM Version 1.2 – 403 – Element is read only • To: SCORM 2004 – 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_data.max_time_allowed	Changed the SCORM dot-notation binding to cmi.max_time_allowed. Changed the data type from CMITimespan to timeinterval (second,10,2). Refer to the SCORM Run-Time Environment book for more information on the timeinterval (second,10,2) data type. Changed the format of the value held by this data model element from HHHH:MM:SS.SS to P[yY][mM][dD][T[hH][mM][s[.s]S]]. Refer to the SCORM Run-Time Environment book for more information on the format of the timeinterval (second,10,2) data type. The data model element initialization procedure was updated to either: • The value stored in the <imsss:attemptAbsoluteDurationLimit> element associated with the <imscp:item> element that references a SCO resource, or • If the <imsss:attemptAbsoluteDurationLimit> element has not been defined., the LMS shall not assume any initialization value. LMS behavior was added for cases where the SCO tries to get the cmi.max_time_allowed and there was no <imsss:attemptAbsoluteDurationLimit> defined. The LMS is responsible for setting the error code to 403 – Data Model Element Not Initialized and returning an empty characterstring. Error code changed for cases where a SCO tries to set cmi.max_time_allowed and the value is not a valid vocabulary token (From 405 - 406) • From: SCORM Version 1.2 – 403 – Element is read only • To: SCORM 2004 – 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_data.time_limit_action	Changed the SCORM dot-notation binding to

	cmi.time_limit_action. Changed the data type from CMIVocabulary to state (exit_message, continue_message, exit_no_message, continue_no_message). Refer to the SCORM Run-Time Environment book for more information on the state data type. The data model element initialization procedure was updated to initialize this element to the value found in the <adlcp:timeLimitAction> element defined on an <imscp:item> that references the SCO resource: If the <adlcp:timeLimitAction> element has not been defined., the LMS shall not assume any initialization value. LMS behavior was added for cases where the SCO tries to get the cmi.time_limit_action and there was no <adlcp:timeLimitAction> defined. The LMS is responsible for setting the error code to 403 – Data Model Element Not Initialized and returning an empty characterstring. Error code changed for cases where a SCO tries to set cmi.time_limit_action and the value is not a valid vocabulary token (From 405 - 406) • From: SCORM Version 1.2 – 403 – Element is read only • To: SCORM 2004 – 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_preference	
cmi.student_preference._children	Changed the SCORM dot-notation binding to cmi.learner_preference._children. Changed the data type from CMIString255 to characterstring (see the SCORM Run-Time Environment book for more information on the characterstring) Removed the requirement of being 255 characters or less. Updated requirement to indicate that there is no requirement on the order of the elements in the comma-separated list. Updated list of children elements returned in the comma-separated list to include new data model element names. Error code changed for cases where a SCO tries to set cmi.learner_preference._children • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is

	Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_preference.audio	Changed the SCORM dot-notation binding to cmi.learner_preference.audio_level. Changed the data type from CMISInteger to real (10,7) in the range of (0..*). Refer to the SCORM Run-Time Environment book for more information on the real (10,7) data type. Based on the data model element data type changes, removed the SCORM Version 1.2 format restriction. Removed the SCORM Version 1.2 initialization of this value to 0. Updated SCORM to indicate that this value can be initialized in various ways dependent on the LMS implementation. LMS behavior was added for cases where the SCO tries to get the cmi.learner_preference.audio_level and no mechanism was provided to initialize this value. The LMS is responsible for setting the error code to 403 – Data Model Element Not Initialized and returning an empty characterstring. Error code changed for cases where a SCO tries to set cmi.learner_preference.audio_level and the value is not a valid real (10,7) number. • From: SCORM Version 1.2 – 405 – Incorrect Data Type • To: SCORM 2004 – 406 Data Model Element Type Mismatch Error code added for cases where a SCO tries to set cmi.learner_preference.audio_level and the value is real (10,7) number; however, the value is not in the required range. • SCORM 2004 – 407 Data Model Element Value Out Of Range Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_preference.language	Changed the SCORM dot-notation binding to cmi.learner_preference.language. Changed the data type from CMIString255 to language_type. Refer to the SCORM Run-Time Environment book for more information on the language_type data type. The value now has a SPM of 250. Based on the data model element data type changes, added

	a specific format for the value held by the data model element. Refer to the SCORM Run-Time Environment book for more information on the language_type data type. Removed the SCORM Version 1.2 initialization of this value to 0. Updated SCORM to indicate that this value can be initialized in various ways dependent on the LMS Iimplementation. Error code changed for cases where a SCO tries to set cmi.learner_preference.language and the value is not a valid real (10,7) number. • From: SCORM Version 1.2 – 405 – Incorrect Data Type • To: SCORM 2004 – 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.student_preference.speed	Changed the SCORM dot-notation binding to cmi.learner_preference.delivery_speed. Changed the data type from CMISInteger to real (10,7) in the range of (0..*). Refer to the SCORM Run-Time Environment book for more information on the real (10,7) data type. Based on the data model element data type changes, removed the SCORM Version 1.2 format restriction (-100 to 100). Removed the SCORM Version 1.2 initialization of this value to an emptystring and require LMSs to initialize this value to "1". Updated SCORM to indicate that this value can be initialized in various ways dependent on the LMS implementation. Error code changed for cases where a SCO tries to set cmi.learner_preference.delivery_speed and the value is not a valid real (10,7) number. • From: SCORM Version 1.2 – 405 – Incorrect Data Type • To: SCORM 2004 – 406 Data Model Element Type Mismatch Error code added for cases where a SCO tries to set cmi.learner_preference.delivery_speed and the value is real (10,7) number, however the value is not in the required range. • SCORM 2004 – 407 Data Model Element Value Out Of Range Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be

	implemented by an LMS.
cmi.student_preference.text	Changed the SCORM dot-notation binding to cmi.learner_preference.audio_captioning. Changed the data type from CMISinteger to state (off, no_change, on). Refer to the SCORM Run-Time Environment book for more information on the state. Error code changed for cases where a SCO tries to set cmi.learner_preference.language and the value defined in the list of state values. • From: SCORM Version 1.2 – 405 – Incorrect Data Type • To: SCORM 2004 – 406 Data Model Element Type Mismatch Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions	Overall general updates: • All interaction data model elements are required to be implemented by an LMS • All interaction data model elements (except for _children and _count) shall be implemented as read/write. • More specific requirements on the format of the correct_response and learner_response data based on the type of interaction • Added requirements of setting the cmi.interactions.n.id first, prior to any other interaction data model elements being set. o Added requirements that the cmi.interactions.n.type be set second, if the correct_response and/or learner_response data model elements will be used by the SCO.
cmi.interactions._children	Changed the data type from CMIString255 to characterstring (Referto the SCORM Run-Time Environment book for more information on the characterstring) Removed the requirement of being 255 characters or less. Updated requirement to indicate that there is no requirement on the order of the elements in the comma-separated list. Updated list of children elements returned in the comma-separated list to include new data model element names. Error code changed for cases where a SCO tries to set cmi.interactions._children

	• From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions._count	Changed the data type from CMIInteger to non-negative integer (Refer to the SCORM Run-Time Environment book for more information on the non-negative integer) Error code changed for cases where a SCO tries to set cmi.interactions._count • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.id	Changed the data type from CMIIdentifier to long_identifier_type (Refer to the SCORM Run-Time Environment book for more information on the long_identifier_type). Added requirements, based on new type, for the identifier to be a Universal Resource Identifier (URI) and recommendation of the value being a Universal Resource Name (URN). A smallest permitted maximum has been defined for the length of the value. The SPM is defined to be 4000 characters. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. The SCO developer needs to ensure that the identifiers are uniquely identifiable within at least the scope of the SCO. Added requirements that the cmi.interactions.n.id shall be the first element sent in a set of interaction data. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a

	GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.id and the data exists but the identifier has not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.id and the value is not a valid long_identifier_type, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Add guidance on the usage of identifiers for the interactions data model element. The identifiers are what uniquely identifies one interaction from another and the identifier should be used to guarantee that correct set of data is being updated (not the position in the array). Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.type	Changed the data type from CMIVocabulary to state (true_false, multiple_choice, fill_in, long_fill_in, matching, performance, sequencing, likert, numeric, other). Refer to the SCORM Run-Time Environment book for more information on the state data type) Changed the binding of the new states to the actual values of the element: • true_false: "true-false" • multiple_choice: "choice" • fill_in: "fill-in" • long_fill_in: "long-fill-in" • matching: "matching" • performance: "performance" • sequencing: "sequencing" • likert: "likert" • numeric: "numeric" • other: "other"

	Note: The quotes are used to delineate the value and are not part of the actual characterstring. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. The SCO developer needs to ensure that the cmi.interactions.n.type is set prior to the cmi.interactions.n.learner_response and/or cmi.interactions.n.correct_response.n.pattern (if those elements will be utilized by the SCO). Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.type and the interactions data record exists but the type does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.type and the value is not a valid state value, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.type prior to the cmi.interactions.n.id element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return “false”. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.

cmi.interactions.n.objectives	
cmi.interactions.n.objectives._count	Changed the data type from CMIInteger to non-negative integer (Refer to the SCORM Run-Time Environment book for more information on the non-negative integer). Error code changed for cases where a SCO tries to set cmi.interactions.n.objectives._count • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.objectives.n.id	Changed the data type from CMIIdentifier to long_identifier_type (Refer to the SCORM Run-Time Environment book for more information on the long_identifier_type). Added requirements, based on new type, for the identifier to be a Universal Resource Identifier (URI) and recommendation of the value being a Universal Resource Name (URN). This is a change of the format from the SCORM Version 1.2. Added an SPM for the identifier (4000 characters). LMSs are required to support at least the SPM for this data model element. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered.

	• If the SCO attempts to set the cmi.interactions.n.objectives.n.id and the value is not a valid long_identifier_type, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.objectives.n.id prior to the cmi.interactions.n.id element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.time	Changed the SCORM dot-notation binding to cmi.interactions.n.timestamp. Changed the data type from CMITime to time (second,10,2). Refer to the SCORM Run-Time Environment book for more information on the time (second,10,2) data type. Changed the format, based on the data type change, from HH:MM:SS.S to YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]]]. Refer to the SCORM Run-Time Environment book for more information on the time (second,10,2) format. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.timestamp and the interactions data record exists but the timestamp does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the

	LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.objectives.n.id and the value is not a valid long_identifier_type, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.timestamp prior to the cmi.interactions.n.id element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.correct_responses	
cmi.interactions.n.correct_responses._count	Changed the data type from CMIInteger to non-negative integer (Refer to the SCORM Run-Time Environment book for more information on the non-negative integer). Error code changed for cases where a SCO tries to set cmi.interactions.n.correct_responses._count • From: SCORM Version 1.2 – 402 Invalid set value, element is keyword. • To: SCORM 2004 - 404 Data Model Element Is Read Only Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.correct_responses.n.pattern	Removed the data type of CMIFeedback. Each type of correct response (base on the cmi.interactions.n.type) now has a detail description of the type and format of the value. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a

	GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.correct_responses.n.pattern and the interactions data record exists but the pattern does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.correct_responses.n.pattern and the value is not a valid according the to the description of the format based on the type of interaction, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.correct_responses.n.pattern prior to the cmi.interactions.n.id and cmi.interactions.n.type element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.weighting	Changed the data type from CMIDecimal to real (10,7). Refer to the SCORM Run-Time Environment book for more information on the real (10,7) data type. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a

	GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.weighting and the interactions data record exists but the weighting does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.weighting and the value is not a valid real (10,7), then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.weighting prior to the cmi.interactions.n.id element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.student_response	Changed the SCORM dot-notation binding to reflect name change within IEEE to: cmi.interactions.n.learner_response. Removed the data type of CMIFeedback. Each type of learner response (base on the cmi.interactions.n.type) now has a detail description of the type and format of the value. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more

	information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.learner_response and the interactions data record exists but the learner_response does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. The SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.learner_response and the value is not a valid according the to the description of the format based on the type of interaction, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.learner_response prior to the cmi.interactions.n.id and cmi.interactions.n.type element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.result	Changed the data type from CMIVocabulary to state (correct, incorrect, unanticipated, neutral, real (10,7)). Refer to the SCORM Run-Time Environment book for more information on the state data type. Changed the binding of the new states to the actual values of the element: • correct: • incorrect: • unanticipated: • neutral: • real (10,7) Note: The quotes are used to delineate the value and are not part of the actual characterstring. The LMS shall implement this element as read/write.

	Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.result and the interactions data record exists but the result does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.result and the value is not a valid state, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.result prior to the cmi.interactions.n.id element has been set. If this scenario happens, the LMS should set the error code to 408 – Data Model Dependency Not Established and return "false". Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.latency	Changed the data type from CMITimespan to timeinterval (second,10,2). Refer to the SCORM Run-Time Environment book for more information on the timeinterval data type. Changed the format of the value held by this data model element from HHHH:MM:SS.SS to P[yY][mM][dD][T[hH][mM][s[.s]S]]. Refer to the SCORM Run-Time Environment book for more information on the format of the timeinterval (second,10,2)

	data type. The LMS shall implement this element as read/write. Removed LMS requirement to set an error code (SCORM Version 1.2 Error Code: 404 – Element is write only) if the SCO invokes a GetValue() request. Added requirements for handling invalid GetValue() requests: • If the SCO invokes a GetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 301 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to retrieve the cmi.interactions.n.latency and the interactions data record exists but the latency does not have a value, the LMS shall set the error code to 403 – Data Model Element Value Not Initialized Added requirements for handling invalid SetValue() requests: • If the SCO invokes a SetValue() request where the index (n) is a number larger than what the LMS is currently maintaining, then the LMS shall set the error code to 351 – General Get Failure. SCORM also recommends how to handle a GetDiagnostic() call occurring to determine more information about the error encountered. • If the SCO attempts to set the cmi.interactions.n.latency and the value is not a valid state, then the LMS shall set the error code to 406 – Data Model Element Type Mismatch. In SCORM Version 1.2 this error code was 405 – Incorrect data type. Added a new error code for the LMS to report if the SCO tries to set the cmi.interactions.n.latency prior to the cmi.interactions.n.id element has been set. If this scenario happens the LMS should set the error code to 408 – Data Model Dependency Not Established and return “false”. Removed the use of error code (401 - Not implemented) from both a GetValue and a SetValue perspective. In SCORM 2004, all elements are required to be implemented by an LMS.
cmi.interactions.n.description	This is a new element defined for SCORM 2004. Information on the specifics of this data model element can be found in the SCORM Run-Time Environment book.
cmi._version	This is a new element defined for SCORM 2004. Information on the specifics of this data model element can be found in the SCORM Run-Time Environment book.

cmi.completion_threshold	This is a new element defined for SCORM 2004. Information on the specifics of this data model element can be found in the SCORM Run-Time Environment book.
cmi.progress_measure	This is a new element defined for SCORM 2004. Information on the specifics of this data model element can be found in the SCORM Run-Time Environment book.

1.5.4.3 Correct Responses and Learner Response Data Type and Format Changes

Several changes have been made to the formats of the characterstrings that are used to represent the `cmi.interactions.n.correct_responses.n.pattern` and `cmi.interactions.learner_response` data model elements. The SCORM Run-Time Environment now defines a specific format for the different types of interactions. Refer to the SCORM Run-Time Environment book for more details on the format of these values.

1.6. SCORM Sequencing and Navigation (SN)

The Sequencing and Navigation book is new to SCORM. This book was added to the SCORM to describe ADL's application profile of the IMS Simple Sequencing Specifications.

1.7. Conversion Tools

ADL has also put together a suite of tools that can be used by content developers in transitioning their SCORM Version 1.2 products to SCORM 2004. The tools being developed include:

- SCORM Version 1.2 to SCORM 2004 Conversion API Wrapper
- SCORM Version 1.2 to SCORM 2004 Generic Run-Time Wrapper
- SCORM Version 1.2 to SCORM 2004 Content Package Manifest XSLT
- SCORM Version 1.2 to SCORM 2004 Meta-data XSLT

These tools are in place to help developers begin to transition their SCORM Version 1.2 content to SCORM 2004. It is anticipated that the ADL Community will begin to produce other tools that will convert content from SCORM Version 1.2 to SCORM 2004. It these tools are developed, it is strongly recommended that these tools be brought to the attention of the ADL Community as a whole in the 3rd Party Tool section of ADLNet.org.

This page intentionally left blank.

APPENDIX A
Acronym Listing

This page intentionally left blank.

Acronym Listing

ADL	Advanced Distributed Learning
AICC	Aviation Industry CBT Committee
API	Application Program Interface
CAM	Content Aggregation Model
CBT	Computer-Based Training
CMI	Computer Managed Instructions
DoD	Department of Defense
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
LMS	Learning Management System
LOM	Learning Objects Metadata
LTSC	Learning Technology Standards Committee
PIF	Package Interchange Format
RFC	Request For Comment
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SN	Sequencing and Navigation
SPM	Smallest Permitted Maximum
URI	Universal Resource Identifier
URL	Universal Resource Locator
XML	eXtensible Markup Language
XSD	XML Schema Definition

APPENDIX B
References

This page intentionally left blank.

References

1. *Sharable Content Object Reference Model (SCORM®) 2004.* January 31, 2004
   SCORM 2004 Overview
   SCORM Content Aggregation Model Version 1.3
   SCORM Run-Time Environment Version 1.3
   SCORM Sequencing and Navigation Version 1.3
   Available at: www.adlnet.org

2. *IEEE 1484.11.1 DRAFT Standard for Learning Technology – Data Model for Content Object Communication.*
   Available at: http://ltsc.ieee.org/

3. *IEEE 1484.11.2 Standard for Learning Technology – ECMAScript Application Programming Interface for Content to Runtime Services Communication.* November 10,2003
   Available at: http://ltsc.ieee.org/

4. *Aviation Industry CBT Committee (AICC) Computer Managed Instruction (CMI) Guidelines for Interoperability Version 3.5.* April 2, 2001
   Available at: http://www.aicc.org/

5. *IEEE 1484.12.1-2002 Learning Object Metadata Standard.*
   Available at: http://www.ieee.org/

6. *IMS Learning Resource Meta-data Specification Version 1.2.1*
   Available at: http://www.imsglobal.org/

7. *IEEE Draft Standard for Extensible Markup Language (XML) Schema Binding for Learning Object Metadata Data Model.*
   Available at: http://ltsc.ieee.org/

8. *IMS Simple Sequencing Behavior and Information Model v1.0 Final Specification, IMS Global Learning Consortium,* March 2003
   Available at: http://www.imsglobal.org/

9. *IETF RFC 1951 DEFLATE Compressed Data Format Specification Version 1.3,* May 1996
   Available at: http://www.ietf.org/

10. *IETF RFC 2396: 1998, Universal Resource Identifiers (URI): General Syntax.*
    Available at: http://www.ietf.org/

11. *ISO/IEC 11404:1996, Information technology – Programming languages, their environments and system software interfaces – Language-independent datatypes.*
    Available at: http://www.iso.ch/

12. *ISO/IEC 10646-1, Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.* Available at: http://www.iso.ch/

13. *IETF RFC 2141: 1997, URN Syntax* Available at: http://www.ietf.org/