SCORM® 2004 3rd EDITION

Sharable Content Object Reference Model

Conformance Requirements

OCTOBER 20, 2006
VERSION 1.0

Advanced Distributed Learning

This page intentionally left blank.

Advanced Distributed Learning (ADL)

SCORM® 2004 3rd Edition

Conformance Requirements (CR)
Version 1.0

Available at ADLNet.gov

For questions and comments visit
Ask The Experts at ADLNet.gov

This page intentionally left blank.

Chief Technical Architect
Philip Dodds

Technical Editor
Schawn E. Thropp

This page intentionally left blank.

COPYRIGHT

DISTRIBUTION

Permission to distribute this document is granted under the following stipulations:

1. The use of this document, its images and examples is for non-commercial, educational or informational purposes only.
2. The document, its images and examples are intact, complete and unmodified. The complete cover page, as well as the COPYRIGHT, DISTRIBUTION and REPRODUCTION sections, are consequently included.

REPRODUCTION

Permission to reproduce this document completely or in part is granted under the following stipulations:

1. The reproduction is for non-commercial, educational or informational purposes only.
2. Appropriate citation of the source document is used as follows:
    a. Source: Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Conformance Requirements Version 1.0, 2006.

For additional information or questions regarding copyright, distribution and reproduction, contact:

ADL Co-Laboratory Hub
1901 North Beauregard Street, Suite 600
Alexandria, Virginia 22311
USA
703-575-2000

This page intentionally left blank.

Table of Contents

List of Tables

Abstract

The SCORM 2004 3rd Edition documentation suite contains a great deal of technical information for a variety of audiences, but product vendors need to know which specific information is critical to making their learning products SCORM 2004 3rd Edition conformant. The ADL Technical Team has collected and structured that information in a concise format that product vendors can reference in the creation of their products.

This document provides a detailed listing of the SCORM conformance requirements as defined in SCORM. Learning Management Systems (LMSs), Sharable Content Objects (SCOs), and/or Content Packages must adhere to these requirements to be recognized as SCORM 2004 3rd Edition conformant. To achieve a conformance label, all conformance requirements for the associated product must be met.

This document is technical by nature and is meant for LMS Vendors, Content Providers, Content Package Creators and those building tools to support these activities.

This page intentionally left blank.

SECTION 1
Introduction

This page intentionally left blank.

1.1. Purpose

The Department of Defense (DoD) established the Advanced Distributed Learning (ADL) Initiative to develop a DoD-wide strategy for using learning and information technologies to modernize education and training. The ADL Initiative has defined high-level requirements (or "-ilities") for learning content, such as interoperability, accessibility, reusability, durability, maintainability and adaptibility, to leverage existing practices, promote the use of technology-based learning and provide a sound economic basis for investment.

SCORM 2004 3rd Edition [1] defines a reference model for sharable learning content objects that meet ADL's high-level requirements. SCORM is an integrated collection of technical specifications that enable conforming Web-based learning products and learning content to interoperate.

It is highly recommended that the reader become familiar with the SCORM 2004 3rd Edition documentation suite before reading this document.

1.2. Scope

This document defines the conformance requirements put forth by SCORM 2004 3rd Edition that must be implemented by Learning Management Systems (LMSs) and/or learning content to be SCORM 2004 3rd Edition conformant.

This document outlines SCORM 2004 3rd Edition conformance requirements for the following products:

1. LMSs
2. Content Packages
    a. SCORM Content Aggregation Content Packages
    b. SCORM Resource Content Packages
3. Sharable Content Objects (SCOs)

1.3. Conformance Requirements Matrices

This section contains conformance matrices that provide a high-level summary of the test subjects that can be tested for SCORM 2004 3rd Edition conformance. These matrices list the conformance labels that describe test subjects that are verified to be conformant to a particular SCORM conformance category, along with a brief description of the requirements that must be implemented by the test subject to achieve the corresponding conformance label. The detailed requirements for conformance are specified in the following sections of this document:

- Section 2: LMS Conformance Requirements

- Section 3: Content Package Conformance Requirements
- Section 4: SCO Conformance Requirements
- Appendix A: Sequencing Conformance Requirements

The Conformance Label is the label used by the SCORM 2004 3rd Edition Conformance Test Suite when the test subject is conformant to the conformance requirements outlined in this document.

The Conformance Category describes the category for which the test subject will be tested for conformance. The SCORM 3rd Edition Conformance Test Suite uses these conformance categories and their corresponding labels throughout the testing process to outline the aspects to which the test subject is conformant. In order to be labeled as SCORM 2004 3rd Edition conformant, the test subject must meet the conformance requirements in each conformance category identified.

1.3.1. LMS Conformance Matrix

Table 1.3.1a defines the high-level conformance requirements that an LMS must adhere to in order to be SCORM 2004 3rd Edition conformant. Each of the Conformance Categories defines the requirements for conformance within that category. The LMS shall adhere to the conformance requirements of each Conformance Category to be SCORM 2004 3rd Edition conformant.

Table 1.3.1a: LMS Conformance Matrix

LMS Conformance Matrix
Conformance Label: • **LMS SCORM 2004 3rd Edition Conformant** The LMS shall adhere to the conformance requirements defined for the following Conformance Categories: • **LMS Run-Time Environment Version 1.0 (LMS RTE 1.0)** • **LMS Content Aggregation Model Version 1.0 (LMS CAM 1.0)** • **LMS Sequencing and Navigation Version 1.0 (LMS SN 1.0)**
Conformance Category: • **LMS RTE 1.0** *Requirements Summary:* • The LMS shall be able to launch an Asset, and • The LMS shall be able to launch a known SCORM 2004 3rd Edition conformant Sharable Content Object (SCO), and • The LMS shall provide and expose an API Instance as a Document Object Model (DOM) object that correctly implements all of the API methods, and • The LMS shall correctly implement support for all SCORM 2004 3rd Edition Run-Time Environment Data Model Elements, and • The LMS shall correctly implement support for all SCORM 2004 3rd Edition Navigation Data Model Elements.
Conformance Category: • **LMS CAM 1.0** *Requirements Summary:* • The LMS shall be able to "import" and process a known SCORM 2004 3rd Edition conformant SCORM Content Aggregation Application Profile Content Package, and • The LMS shall correctly initialize SCORM 2004 3rd Edition Run-Time Environment Data Model elements based on information supplied in a Content Package Manifest
Conformance Category: • **LMS SN 1.0** *Requirements Summary:* • The LMS shall correctly implement all of the sequencing behaviors defined by the pseudo-code included in the SCORM 2004 3rd Edition Sequencing and Navigation (SN) Version 1.0, and • The LMS shall correctly implement support for all SCORM 2004 3rd Edition Navigation Data Model Elements, and • The LMS shall correctly implement support for Navigation User Interface requirements.

1.3.2. Content Package Conformance Matrix

Table 1.3.2a defines the high-level conformance requirements that a Content Package must adhere to in order to be SCORM 2004 3rd Edition conformant. Each of the Conformance Categories defines the requirements for conformance within that category. The Content Package shall adhere to the conformance requirements of each Conformance Category to be SCORM 2004 3rd Edition conformant.

Table 1.3.2a: Content Package Conformance Matrix

Content Package Conformance Matrix
Conformance Label: • **CP SCORM 2004 3rd Edition Conformant** The Content Package shall adhere to the conformance requirements defined for the following Conformance Categories: • **Content Package Content Aggregation Model Version 1.0 (CP CAM 1.0)** • **Content Package Run-Time Environment Version 1.0 (CP RTE 1.0)**
Conformance Category: • **CP CAM 1.0** *Requirements Summary:* • The Content Package shall conform to the requirements defined for the Content Package, and • If the Content Package is a SCORM 2004 3rd Edition Content Aggregation Package Application Profile, then the Manifest shall conform to the Content Aggregation Package Application Profile Manifest requirements, and • If the Content Package is a SCORM 2004 3rd Edition Content Aggregation Package Application Profile and the Manifest contains SCORM Sequencing information, then the sequencing extensions in the Manifest shall conform to the SCORM 2004 3rd Edition Sequencing Extension requirements, and • If the Content Package is a SCORM 2004 3rd Edition Content Aggregation Package Application Profile and the Manifest contains SCORM Navigation/Presentation information, then the navigation/presentation extensions in the Manifest shall conform to the SCORM 2004 3rd Edition Navigation/Presentation Extension requirements, and • If the Content Package is a SCORM 2004 3rd Edition Resource Package Application Profile, then the Manifest shall conform to the Resource Package Application Profile Manifest requirements, and • If the Content Package Manifest contains metadata, then the metedata shall be well-formed and valid according to the respective Controlling Document (e.g., XSD, DTD).
Conformance Category: • **CP RTE 1.0** *Requirements Summary:* • The Content Package shall contain at least one Sharable Content Object (SCO) resource or Asset resource, and • All SCO resources identified in the Manifest shall conform to the SCO conformance requirements.

1.3.3. SCO Conformance Matrix

Table 1.3.3a defines the high-level conformance requirements that a SCO must adhere to in order to be SCORM 2004 3rd Edition conformant. Each of the Conformance Categories defines the requirements for conformance within that category. The SCO shall adhere to the conformance requirements of each Conformance Category to be SCORM 2004 3rd Edition conformant.

Table 1.3.3a: SCO Conformance Matrix

SCO Conformance Matrix
Conformance Label: • **SCO SCORM 2004 3rd Edition Conformant** The SCO shall adhere to the conformance requirements defined for the following Conformance Categories: • **SCO Run-Time Environment Version 1.0 (SCO RTE 1.0)**
Conformance Category: • **SCO RTE 1.0** *Requirements Summary:* • The SCO shall search for and find an API Instance named API_1484_11 as a Document Object Model (DOM) object as defined in the SCORM 3rd Edition Run-Time Environment Version 1.0, and • The SCO shall successfully invoke, at a minimum, the Initialize("") and Terminate("") API methods, and • If used, the SCO shall successfully invoke the Data Transfer Methods, and • If used, the SCO shall successfully invoke the Support Methods, and • If using the Data Transfer Methods, the SCO shall ensure that all SCORM 2004 3rd Edtiion Run-Time Environment Data Model elements used in the method calls adhere to the requirements of those elements.

1.4. Compliant vs. Conformant vs. Certified

The terms “compliant,” “conformant” and “certified” are used throughout the ADL Community in different contexts. ADL avoids using the term “SCORM compliant” in favor of “SCORM conformant,” even though e-learning experts often use the terms “SCORM conformant” and “SCORM compliant” interchangeably. The term “conformant” should be used when describing a product that follows the SCORM 2004 3rd Edition specifications.

A “SCORM conformant” product is defined as a product that has passed the SCORM Conformance Test Suite (Self Test), which indicates that the product conforms to the latest version of SCORM specifications as outlined in the SCORM Conformance Requirements. Anyone whose products pass the SCORM Conformance Test Suite can claim to be SCORM conformant.

A “SCORM certified” product has been independently tested by one of the ADL Certification Testing Centers and upon passing the SCORM Conformance Test Suite, becomes “ADL certified.” ADL certification assures consumers of distributed learning content and systems that certified products have successfully implemented SCORM specifications.

From a technical standpoint, an “ADL certified” product is no different than a “SCORM conformant” product. The key difference is that “ADL certified” products have been independently tested not by the LMS vendors or content creators themselves, but by the independent ADL Certification Testing Centers.

1.5. Conformance Requirements Overview

Each of the conformance requirements tables that are defined in the upcoming sections has the following format:

Table 1.5a: Sample Conformance Requirement Table Format

REQ ID	Requirement
REQ_<*id*>	<Text of the conformance requirement>.

Each conformance requirement is given a unique identifier (REQ_<id>). The <id> portion is a unique number used to identify and label the conformance requirement. The REQ_<id> does not necessarily imply a specific sequence to the requirements. The Requirement column outlines the specific conformance requirement.

SECTION 2
LMS Conformance Requirements

This page intentionally left blank.

2.1. LMS Conformance Requirements

This section describes the detailed requirements that must be implemented by an LMS to be SCORM 2004 3rd Edition conformant. In order to be SCORM 2004 3rd Edition conformant, an LMS is required to support various aspects of the SCORM Run-Time Environment (RTE) [1], the SCORM Content Aggregation Model (CAM) [1] and the SCORM Sequencing and Navigation (SN) [1].

The conformance requirements for an LMS are divided into the following sections to address different aspects of SCORM individually. They are as follows:

- Section 2.1.1: Launch Conformance Requirements
- Section 2.1.2: API Implementation Conformance Requirements
- Section 2.1.3: Run-Time Environment Data Model Conformance Requirements
- Section 2.1.4: Run-Time Environment Data Model Data Type Conformance Requirements
- Section 2.1.5: Navigation Data Model Conformance Requirements
- Section 2.1.6: Sequencing Conformance Requirements

The purpose of the LMS Conformance Test is to verify that an LMS implements the conformance requirements as outlined in this section. The LMS Conformance Test is designed to test several conformance categories:

- Conformance Category 1: **LMS RTE 1.0** – The LMS is conformant to the requirements defined in the SCORM 3rd Edition Run-Time Environment Version 1.0.

- Conformance Category 2: **LMS CAM 1.0** – The LMS is conformant to the requirements defined in the SCORM 3rd Edition Content Aggregation Model Version 1.0.

- Conformance Category 3: **LMS SN 1.0** – The LMS is conformant to the requirements defined in the SCORM 3rd Edition Sequencing and Navigation Version 1.0.

In order to test these conformance categories, the LMS Conformance Test uses a set of SCORM Content Aggregation Packages that are to be imported into the LMS. The content packages, manifests, sequencing rules and learning resources (SCOs and Assets) that comprise the content packages exercise the various aspects of the LMS' implementation of SCORM 2004 3rd Edition.

Based on the satisfaction of the LMS Conformance Requirements defined for the above conformance categories, the LMS may or may not be found to be **SCORM 2004 3rd Edition LMS Conformant**.

A key requirement defined by the LMS CAM 1.0 Conformance Category is that the LMS shall provide the capability to import a SCORM 3rd Edition Content Aggregation

Packages using a SCORM Package Interchange File (PIF). It is necessary for the LMS to do this with content packages in order to attain conformance to any of the conformance categories.

During the execution of the set of SCORM 3rd Edition Content Aggregation Packages, the LMS Conformance Test will test an LMS' conformance with its ability to:

- launch a learning resource (SCO or Asset)
- implement the SCORM 3rd Edition Run-Time Environment API
- implement the SCORM 3rd Edition Run-Time Environment Data Model
- implement the SCORM 3rd Edition Navigation Data Model
- adhere to the SCORM 3rd Edition Content Aggregation Model
- implement the SCORM 3rd Edition Sequencing Behaviors

2.1.1. Launch Conformance Requirements

It is the responsibility of the LMS to determine which learning resource (Asset or SCO) is to be launched to the learner. The LMS must determine which Asset or SCO to launch to the learner based on the defined organization, and possibly the sequencing information defined in the content package manifest (`imsmanifest.xml`). The overall launching of learning resources could be different from learner to learner based on the following factors:

- Sequencing information defined for the learning activities for which the learning resource is associated.
- Learner's performance within the overall context of the content aggregation for which the learning resource is contained in.
- The state of the learning experience in relation to the content and the content aggregation for which it is contained in.
- The LMS shall adhere to the requirements defined in Table 2.1.1a to be considered conformant to the LMS RTE 1.0 conformance category.

Table 2.1.1a: LMS Launch Conformance Requirements

REQ ID	Requirement
REQ_22	The LMS shall launch the learning resources defined in the Content Package Manifest (`imsmanifest.xml`) based on the <**resource**> referenced by a leaf <**item**> that is found in the content organization (<**organization**>).
REQ_22.1	The LMS shall be able to launch a SCORM 2004 3rd Edition conformant Sharable Content Object (SCO). SCOs are identified in an `imsmanifest.xml` as a <**resource**> with an attribute of **adlcp:scormType**="sco".
REQ_22.2	The LMS shall be able to launch a SCORM Asset. Assets are identified in an `imsmanifest.xml` as a <**resource**> with an attribute of **adlcp:scormType**="asset".
REQ_23	The LMS shall launch learning resources using the Hypertext Transfer Protocol

REQ ID	Requirement
	(HTTP) protocol.
REQ_24	The LMS shall launch a learning resource in a Document Object Model (DOM) frameset child window, or new browser (DOM) window relative to the LMS' controlling browser (DOM) window.
REQ_25	The LMS shall only launch one SCO at a time. From the LMS' perspective there may only be one "currently executing" SCO.

2.1.2. API Implementation Conformance Requirements

The SCORM Run-Time Environment API provides a consistent means by which SCOs can communicate and exchange data with LMSs. The LMS is required to implement an API Instance that supports the methods defined by the SCORM Run-Time Environment.

The LMS shall adhere to the requirements defined in Table 2.1.2a to be considered conformant to the LMS RTE 1.0 conformance category.

Table 2.1.2a: LMS API Implementation Conformance Requirements

REQ ID	Requirement
REQ_1	The LMS shall implement the API so that it can be invoked via ECMAScript (JavaScript).
REQ_1.1	The LMS' API implementation shall allow its methods to be invoked using ECMAScript (JavaScript).
REQ_1.2	The LMS shall implement all API implementation method parameters as ECMAScript characterstrings.
REQ_1.3	The LMS shall implement all API implementation method return values as ECMAScript characterstrings.
REQ_1.4	The LMS' API implementation method names shall be case-sensitive.
REQ_1.5	The LMS' API method parameters and return values that represent integers, real numbers, durations, and times shall be encoded as they would be by the ECMAScript-to-string cast conversion.
REQ_2	The LMS shall implement the API so that is accessible via a DOM object.
REQ_2.1	The LMS' API implementation shall be made accessible as an instantiated object in the DOM environment of the SCO.
REQ_2.2	The LMS' API implementation shall be instantiated in any of the following DOM elements:
REQ_2.2.1	The LMS' API Implementation can be located in any window in the chain of parents of the window within which the SCO is launched, up to and including the top window of the Web browser.
REQ_2.2.2	The LMS' API Implementation can be located in the opener window of the window within which the SCO is launched.
REQ_2.2.3	The LMS' API Implementation can be located in any window in the chain of parents of the opener window within which the SCO is launched, if any exist, up to and including the top window of the Web browser.
REQ_2.3	The LMS' API instance shall be instantiated before the SCO is launched.

REQ ID	Requirement
REQ_2.4	The API instance, provided by an LMS, shall be responsible for maintaining a communication session with 1 and only 1 SCO during the duration of the SCO.
REQ_2.5	The LMS' API instance shall be a DOM object named `API_1484_11`.
REQ_2.6	The LMS' API instance (DOM object) shall have an attribute named `version`.
REQ_2.6.1	The LMS' API instance (DOM object) version attribute shall have the first three characters of its value be: `1.0`.
REQ_2.6.2	If there are more than three characters in the value of the API instance's version attribute the fourth character shall be: • `.` ("period" - the defined separator). Any following characters are implementation defined.
REQ_3	An LMS' API Implementation States shall be implemented as follows:
REQ_3.1	When an LMS' API is instantiated it shall have a starting error code of `0`.
REQ_3.2	If a non-empty characterstring parameter is passed to a method that requires an empty characterstring parameter (e.g. Initialize, Terminate, Commit), error code `201` shall be returned by the LMS.
REQ_4	The LMS shall adhere to the following **Initialize()** API method requirements:
REQ_4.1	The LMS shall implement the **Initialize()** API method with the following signature: **return_status = Initialize("")**.
REQ_4.1.1	The parameter passed into the **Initialize()** API method shall be an empty characterstring ("").
REQ_4.2	If the communication session has not been initialized and the **Initialize()** API method is invoked and fails, the LMS shall set the error code to General Initialization Failure (102) and return `false`.
REQ_4.3	If the communication session has been initialized and the **Initialize()** API method is invoked again, the LMS shall set the error code to Already Initialized (103) and return `false`.
REQ_4.4	If the communication session has been terminated and the **Initialize()** API method is invoked, the LMS shall set the error code to Content Instance Terminated (104) and return `false`.
REQ_4.5	If the communication session has not been initialized and the **Initialize()** API method is successful, the LMS shall set the error code to No Error (0) and return `true`.
REQ_5	The LMS shall adhere to the following **Terminate()** API method requirements:
REQ_5.1	The LMS shall implement the **Terminate()** API method with the following signature: **return_status = Terminate("")**.
REQ_5.1.1	The parameter passed into the **Terminate()** API method shall be an empty characterstring ("").
REQ_5.2	If the communication session has been initialized and the **Terminate()** API method is successful, the LMS shall set the error code to No Error (0) and return `true`.
REQ_5.2.1	If the communication session has been initialized and the **Terminate()** API method is successful, the LMS shall persist any data set by the SCO (equivalent to an implicit Commit method call).
REQ_5.2.1.1	If the persistence of data fails, the LMS shall set the error code to General Commit Failure (391) and return `false`.
REQ_5.3	If the communication session has been initialized and the **Terminate()** API method fails, the LMS shall set the error code to General Termination Failure (111) and return

REQ ID	Requirement
	`false`.
REQ_5.4	If the communication session has not been initialized and the **Terminate()** API method is invoked, the LMS shall set the error code to Termination Before Initialization (112) and return `false`.
REQ_5.5	If the communication session has been terminated and the **Terminate()** API method is invoked, the LMS shall set the error code to Termination After Termination (113) and return `false`.
REQ_6	The LMS shall adhere to the following **GetValue()** API method requirements:
REQ_6.1	The LMS shall implement the **GetValue()** API method with the following signature: **return_value = GetValue(parameter)**.
REQ_6.1.1	The `parameter` shall be a data model element name for which a value is to be retrieved.
REQ_6.2	If the communication session has been initialized and the **GetValue()** API method is invoked successfully, the LMS shall set the error code to No Error (0) and return the requested data.
REQ_6.3	If the communication state has been initialized and the **GetValue()** API method is invoked where the `parameter` is not recognized by the LMS, the LMS shall set the error code to Undefined Data Model Element (401) and return an empty characterstring ("").
REQ_6.4	If the communication state has been initialized and the **GetValue()** API method is invoked where the `parameter` is recognized but not implemented by the LMS, the LMS shall set the error code to Unimplemented Data Model Element (402) and return an empty characterstring ("").
REQ_6.5	If the communication state has been initialized and the **GetValue()** API method is invoked where the request is for a data model element that has not been initialized with a value, the LMS shall set the error code to Data Model Element Value Not Initialized (403) and return an empty characterstring ("").
REQ_6.6	If the communication state has been initialized and the **GetValue()** API method is invoked where the request is for a data model element that is write-only, the LMS shall set the error code to Data Model Element Is Write Only (405) and return an empty characterstring ("").
REQ_6.7	If the communication state has been initialized and the **GetValue()** API method is invoked unsuccessfully and no specific error condition exists, the LMS shall set the error code to General Get Failure (301) and return an empty characterstring (""). **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_6.8	If the communication state has not been initialized and the **GetValue()** API method is invoked, the LMS shall set the error code to Retrieve Data Before Initialization (122) and return an empty characterstring ("").
REQ_6.9	If the communication state has been terminated and the **GetValue()** API method is invoked, the LMS shall set the error code to Retrieve Data After Termination (123) and return an empty characterstring ("").
REQ_6.10	If the communication session has been initialized and the **GetValue()** API method is invoked where `parameter` is not specified (i.e., specified by an empty characterstring – “”), then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring (“”). **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for

REQ ID	Requirement
	using this error condition.
REQ_6.11	If the communication state has been initialized and the **GetValue()** API method is invoked requesting the children of a data model element that does not have children, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring (""). **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_6.12	If the communication state has been initialized and the **GetValue()** API method is invoked requesting the number of entries (i.e., _count) currently stored in a data model element that is not an array, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring (""). **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_7	The LMS shall adhere to the following **SetValue()** API method requirements:
REQ_7.1	The LMS shall implement the **SetValue()** API method with the following signature: **return_status = SetValue(parameter_1, parameter_2)**.
REQ_7.1.1	`parameter_1` shall be a characterstring value representing a complete identification of a data model element within a data model.
REQ_7.1.2	`parameter_2` shall be a characterstring value representing the value to which the contents of `parameter_1` are to be set.
REQ_7.2	If the communication session has been initialized and the **SetValue()** API method is successful, the LMS shall set the error code to No Error (0) and return `true`.
REQ_7.3	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_1` is not recognized by the LMS, the LMS shall set the error code to Undefined Data Model Element (401) and return `false`.
REQ_7.4	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_1` is recognized but not implemented by the LMS, the LMS shall set the error code to Unimplemented Data Model Element (402) and return `false`.
REQ_7.5	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_1` is read-only data model element, the LMS shall set the error code to Data Model Element Is Read Only (404) and return `false`.
REQ_7.6	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_2` does not match the datatype required by `parameter_1`, the LMS shall set the error code to Data Model Element Type Mismatch (406) and return `false`.
REQ_7.7	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_2` matches the datatype of `parameter_1` but the value was not in the specified range of values required for `parameter_1`, the LMS shall set the error code to Data Model Element Value Out Of Range (407) and return `false`.
REQ_7.8	If the communication session has been initialized and the **SetValue()** API method is invoked where relevant dependencies are not in place (See the Run-Time Environment Data Model conformance requirements for additional details), the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`.
REQ_7.9	If the communication session has not been initialized and the **SetValue()** API method is invoked, the LMS shall set the error code to Store Data Before Initialization (132) and return `false`.

REQ ID	Requirement
REQ_7.10	If the communication session has been terminated and the **SetValue()** API method is invoked, the LMS shall set the error code to Store Data After Termination (133) and return `false`.
REQ_7.11	If the communication state has been initialized and the **SetValue()** API method is invoked unsuccessfully and no specific error condition exists, the LMS shall set the error code to General Set Failure (351) and return `false`. **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_7.12	If the communication state has been initialized and the **SetValue()** API method is invoked on a data model element that would cause a data model record "to-be" created (those data model elements that are members of a collection) and the call is unsuccessful, then the LMS shall not create the data model record and the LMS shall not increase the size of the containing collection. In these cases, the LMS shall return `false` and set the appropriate API error code.
REQ_7.13	If the communication session has been initialized and the **SetValue()** API method is invoked where `parameter_1` is not specified (i.e., specified by an empty characterstring – ""), then the LMS shall set the error code to General Set Failure (351) and return `false`. **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_7.14	If the communication session has been initialized and the **SetValue()** API method is invoked where a SCO attempted to set a new value (i.e., not replace an existing value) in an array where the index number used (n) is not the next available position in the array, then the LMS shall set the error code to General Set Failure (351) and return `false`. **ADL NOTE:** See the SCORM Run-Time Environment for more specific situations for using this error condition.
REQ_7.15	If the communication session has been initialized and the **SetValue()** API method is invoked where a SCO attempted to set a data model element and the defined SPM for that data model element is exceeded, then the LMS shall set at least the defined SPM for that data model element, set the error code to No Error (0) and return `true`. **ADL NOTE:** See the SCORM Run-Time Environment for more details on the SPMs defined for the data model elements.
REQ_8	The LMS shall adhere to the following **Commit()** API method requirements:
REQ_8.1	The LMS shall implement the **Commit()** API method with the following signature: **return_status = Commit("")**.
REQ_8.1.1	The parameter passed into the **Commit()** API method shall be an empty characterstring ("").
REQ_8.2	If the communication session has been initialized and the **Commit()** API method is successful, the LMS shall set the error code to No Error (0) and return `true`.
REQ_8.2.1	During a **Commit()** API method call, the LMS shall persist any data that was set, and has not been persisted since the last successful call to **Commit("")** or **Initialize("")**, whichever occurred most recently.
REQ_8.3	If the communication state has been initialized and the **Commit()** API method fails, the LMS shall set the error code to General Commit Failure (391) and return `false`
REQ_8.4	If the communication state has not been initialized and the **Commit()** API method is invoked, the LMS shall set the error code to Commit Before Initialization (142) and return `false`.

REQ ID	Requirement
REQ_8.5	If the communication state has been terminated and the **Commit()** API method is invoked, the LMS shall set the error code to Commit After Termination (143) and return `false`.
REQ_9	The LMS shall adhere to the following **GetLastError()** API method requirements:
REQ_9.1	The LMS shall implement the **GetLastError()** API method with the following signature: **return_status = GetLastError()**.
REQ_9.2	Regardless of the API's state (Not Initialized, Running, Terminated) the LMS shall return a characterstring (convertible to an integer in the range from 0 to 65536 inclusive) representing the error code of the last error encountered.
REQ_9.3	If an LMS receives subsequent calls to **GetLastError()** with no other interceding calls to any other API methods, then the LMS shall return the same error code (returned previously).
REQ_10	The LMS shall adhere to the following **GetErrorString()** API method requirements:
REQ_10.1	The LMS shall implement the **GetErrorString()** API method with the following signature: **return_text = GetErrorString(parameter)**.
REQ_10.2	The `parameter` passed into the LMS' **GetErrorString()** API method shall be the characterstring representing an error code.
REQ_10.3	Regardless of the API's state (Not Initialized, Running, Terminated), the LMS shall return any text associated with the `parameter` when the specified method `parameter` value is known.
REQ_10.3.1	The characterstring returned shall have a maximum length of 255 characters.
REQ_10.4	If the `parameter` passed to **GetErrorString()** is not known, the LMS shall return an empty characterstring ("").
REQ_11	The LMS shall adhere to the following **GetDiagnostic()** API method requirements:
REQ_11.1	The LMS shall implement the **GetDiagnostic()** API method with the following signature: **diagnostic_text = GetDiagnostic(parameter)**.
REQ_11.2	Regardless of the API's state (Not Initialized, Running, Terminated), the LMS shall return diagnostic information for the specified method `parameter` value when the `parameter` value passed to the GetDiagnostic API method is known.
REQ_11.2.1	The textual characterstring returned shall have a maximum length of 255 characters.
REQ_11.3	Regardless of the API's state (Not Initialized, Running, Terminated), the LMS shall return an empty characterstring ("") when the `parameter` value passed to the GetDiagnostic API method is not known.
REQ_11.4	The `parameter` passed into the LMS' **GetDiagnostic()** API method shall be a characterstring.

2.1.3. Run-Time Environment Data Model Conformance Requirements

The SCORM Run-Time Environment Data Model contains a set of data model elements that can be tracked by the SCO with an LMS during the run-time execution of the SCO. The data model describes the information that is exchanged between SCOs and LMSs via

the API. Future versions of SCORM may include additional or alternative communication mechanisms and/or data models. For this reason, the following data model conformance requirements are written such that they can stand independently from the communication mechanism.

Conformance to the API Implementation Conformance Requirements (refer to Section 2.1.2) is a prerequisite for conformance to the SCORM Run-Time Environment Data Model conformance requirements, since there is currently only one communication mechanism supported by SCORM.

LMS are required to implement all of the SCORM Run-Time Environment Data Model elements. The following list provides a description of the key terms that are used in the requirements tables in this section to describe the Run-time Environment Data Model conformance requirements:

- **read-only** – The LMS shall implement this element such that a SCO may only retrieve (read) the value using the GetValue() API method. If the SCO attempts to store (write) a value for this element using the SetValue() API method, the LMS shall behave according to the API Implementation Conformance Requirements (refer to Section 2.1.2).

  For read-only data model elements, SCOs may only invoke the GetValue() API method .

- **write-only** – The LMS shall implement this element such that a SCO may only store (write) the value using the SetValue() API method. If the SCO attempts to retrieve (read) a value for this element using the GetValue() API method, the LMS shall behave according to the API Implementation Conformance Requirements (refer to Section 2.1.2).

  For write-only data model elements, SCOs may only invoke the SetValue() API method.

- **read/write** – The LMS shall implement this element such that a SCO may retrieve (read) the value using the GetValue() API method or store (write) the value using the SetValue() API method.

  For read/write data model elements, SCOs may invoke the SetValue() or GetValue() API method.

Section 2.1.4 Run-Time Environment Data Model Data Type Conformance Requirements defines additional requirements about the defined datatypes for each of the data model elements. LMSs shall adhere to the requirements defined in this section.

The LMS shall adhere to the requirements defined in Table 2.1.3a to be considered conformant to the LMS RTE 1.0 conformance category.

Table 2.1.3a: LMS Run-Time Environment Data Model Conformance Requirements

REQ ID	Requirement
REQ_55	The LMS shall implement the **cmi._version** data model element.
REQ_55.1	The LMS shall implement the **cmi._version** data model element as read-only.
REQ_55.2	The LMS shall implement the **cmi._version** data model element as a characterstring.
REQ_55.3	If a SCO invokes a GetValue() request to retrieve the **cmi._version**, then the LMS shall return the characterstring `1.0`.
REQ_57	The LMS shall support at least the Smallest Permitted Maximum (SPM) of 250 comments from learner.
REQ_57.1	The LMS shall implement the **cmi.comments_from_learner._children** data model element.
REQ_57.1.1	The LMS shall implement the **cmi.comments_from_learner._children** data model element as a read-only.
REQ_57.1.2	The LMS shall implement the **cmi.comments_from_learner._children** data model element as a characterstring.
REQ_57.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.comments_from_learner._children**, then the LMS shall return a comma-separated list of all **cmi.comments_from_learner** child data model elements: • `comment` • `location` • `timestamp` **ADL NOTE:** The order of these values is not significant.
REQ_57.2	The LMS shall implement the **cmi.comments_from_learner._count** data model element.
REQ_57.2.1	The LMS shall implement the **cmi.comments_from_learner._count** data model element as read-only.
REQ_57.2.2	The LMS shall implement the **cmi.comments_from_learner._count** data model element as a non-negative integer.
REQ_57.2.3	If a SCO invokes a GetValue() request to retrieve the **cmi.comments_from_learner._count**, then the LMS shall return a characterstring representing the number of learner comments currently stored by the LMS.
REQ_57.3	The LMS shall implement the **cmi.comments_from_learner.n.comment** data model element.
REQ_57.3.1	The LMS shall implement the **cmi.comments_from_learner.n.comment** data model as read/write.
REQ_57.3.2	The LMS shall implement the **cmi.comments_from_learner.n.comment** data model element as a characterstring with an SPM of 4000 characters.
REQ_57.3.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_learner.n.comment** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").

REQ ID	Requirement
REQ_57.3.4	If the SCO invokes a SetValue() request on the **cmi.comments_from_learner.n.comment** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_57.4	The LMS shall implement the **cmi.comments_from_learner.n.location** data model element.
REQ_57.4.1	The LMS shall implement the **cmi.comments_from_learner.n.location** data model element as read/write.
REQ_57.4.2	The LMS shall implement the **cmi.comments_from_learner.n.location** data model element as a characterstring with an SPM of 250 characters.
REQ_57.4.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_learner.n.location** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring (“”).
REQ_57.4.4	If the SCO invokes a SetValue() request on the **cmi.comments_from_learner.n.location** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_57.5	The LMS shall implement the **cmi.comments_from_learner.n.timestamp** data model element.
REQ_57.5.1	The LMS shall implement the **cmi.comments_from_learner.n.timestamp** data model element as read/write.
REQ_57.5.2	The LMS shall implement the **cmi.comments_from_learner.n.timestamp** data model element as a time (second,10,0). This value shall be accurate to one second.
REQ_57.5.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_learner.n.timestamp** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_57.5.4	If the SCO invokes a SetValue() request on the **cmi.comments_from_learner.n.timestamp** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_learner** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_58	The LMS shall support at least the SPM of 100 comments from the LMS.
REQ_58.1	The LMS shall implement the **cmi.comments_from_lms._children** data model element.
REQ_58.1.1	The LMS shall implement the **cmi.comments_from_lms._children** data model element as read-only.
REQ_58.1.2	The LMS shall implement the **cmi.comments_from_lms._children** data model element as a characterstring.

REQ ID	Requirement
REQ_58.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.comments_from_lms._children**, then the LMS shall return a comma-separated list of all **cmi.comments_from_lms** child data model elements: • `comment` • `location` • `timestamp` **ADL NOTE:** The order of these values is not significant.
REQ_58.2	The LMS shall implement the **cmi.comments_from_lms._count** data model element.
REQ_58.2.1	The LMS shall implement the **cmi.comments_from_lms._count** data model element as read-only.
REQ_58.2.2	The LMS shall implement the **cmi.comments_from_lms._count** data model element as non-negative integer.
REQ_58.2.3	If a SCO invokes a GetValue() request to retrieve the **cmi.comments_from_lms._count**, then the LMS shall return the number of **cmi.comments_from_lms** currently stored by the LMS.
REQ_58.3	The LMS shall implement the **cmi.comments_from_lms.n.comment** data model element.
REQ_58.3.1	The LMS shall implement the **cmi.comments_from_lms.n.comment** data model element as read-only.
REQ_58.3.2	The LMS shall implement the **cmi.comments_from_lms.n.comment** data model element as a localized_string_type with an SPM of 4000 characters.
REQ_58.3.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_lms.n.comment** data model element where the index (n) provided is a number that is greater than the current number of **cmi.comments_from_lms** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_58.4	The LMS shall implement the **cmi.comments_from_lms.n.location** data model element.
REQ_58.4.1	The LMS shall implement the **cmi.comments_from_lms.n.location** data model element as read-only.
REQ_58.4.2	The LMS shall implement the **cmi.comments_from_lms.n.location** data model element as a characterstring with an SPM of 250 characters.
REQ_58.4.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_lms.n.location** data model element where the index (n) provided is a number greater than the current number of **cmi.comments_from_lms** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_58.5	The LMS shall implement the **cmi.comments_from_lms.n.timestamp** data model element.
REQ_58.5.1	The LMS shall implement the **cmi.comments_from_lms.n.timestamp** data model element as read-only.

REQ ID	Requirement
REQ_58.5.2	The LMS shall implement the **cmi.comments_from_lms.n.timestamp** data model element as a time (second,10,0). This value shall be accurate to one second.
REQ_58.5.3	If the SCO invokes a GetValue() request on the **cmi.comments_from_lms.n.timestamp** data model element where the index (n) provided is a number greater than the current number of **cmi.comments_from_lms** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_59	The LMS shall implement the **cmi.completion_status** data model element.
REQ_59.1	The LMS shall implement the **cmi.completion_status** data model element as read/write.
REQ_59.2	The LMS shall implement the **cmi.completion_status** data model element as a state consisting of the following vocabulary tokens: • `completed` • `incomplete` • `not attempted` • `unknown`
REQ_59.3	The LMS shall initialize the value of the **cmi.completion_status** data model element to the default value of `unknown`.
REQ_59.4	The LMS shall implement the **cmi.completion_status** such that it has the following effect on sequencing behaviors:
REQ_59.4.1	If the SCO or LMS sets the **cmi.completion_status** of the SCO to `unknown`, then the LMS' sequencing implementation shall behave as if the Attempt Progress Status for the learning activity associated with the SCO is `false`.
REQ_59.4.2	If the SCO or LMS sets the **cmi.completion_status** of the SCO to `completed`, then the LMS' sequencing implementation shall behave as if the Attempt Progress Status for the learning activity associated with the SCO is `true`, and the Attempt Completion Status for the learning activity associated with the SCO is `true`.
REQ_59.4.3	If the SCO or LMS sets the **cmi.completion_status** of the SCO to `incomplete`, then the LMS' sequencing implementation shall behave as if the Attempt Progress Status for the learning activity associated with the SCO is `true`, and the Attempt Completion Status for the learning activity associated with the SCO is `false`.
REQ_59.4.4	If the SCO or LMS sets **cmi.completion_status** of the SCO to `not attempted`, then the LMS' sequencing implementation shall behave as if the Attempt Progress Status for the learning activity associated with the SCO shall be `true` and the Attempt Completion Status for the learning activity associated with the SCO shall be `false`.
REQ_59.5	The LMS shall evaluate the value of the **cmi.completion_status** data model element and return the result in the response to a `GetValue()` request according to the following requirements:
REQ_59.5.1	If a **cmi.completion_threshold** is defined for the SCO and the **cmi.progress_measure** data model element's value is set by the SCO and the value is less than the **cmi.completion_threshold** data model element's value, then the LMS shall evaluate and return the value of `incomplete`.

REQ ID	Requirement
REQ_59.5.2	If a **cmi.completion_threshold** is defined for the SCO and the **cmi.progress_measure** data model element's value is set by the SCO and the value is greater than or equal to the **cmi.completion_threshold** data model element's value, then the LMS shall evaluate and return the value of `completed`.
REQ_59.5.3	If a **cmi.completion_threshold** is defined for the SCO and the **cmi.progress_measure** data model element is never set by the SCO, the LMS shall evaluate and return the value of `unknown`.
REQ_59.5.4	If no **cmi.completion_threshold** is defined for the SCO the LMS shall rely on the value set for the **cmi.completion_status** data model element by the SCO and return that value. If no value was set by the SCO for the **cmi.completion_status** data model element then the LMS shall return `unknown`.
REQ_60	The LMS shall implement the **cmi.completion_threshold** data model element.
REQ_60.1	The LMS shall implement the **cmi.completion_threshold** data model element as read-only.
REQ_60.2	The LMS shall implement the **cmi.completion_threshold** data model element as real (10,7). The data model element shall support a range between (0.0 .. 1.0).
REQ_60.3	The LMS shall initialize this data model element using the SCORM 2004 3rd Edition Content Packaging Extensions Version 1.0 namespace element **<adlcp:completionThreshold>**. If an **<adlcp:completionThreshold>** element does not exist as a child element of the **<imscp:item>** element (associated with the SCO resource), then the element shall remain uninitialized. **NOTE:** In this case uninitialized is defined to indicate that no value should be assigned to cmi.completion_threshold. In this case, if a SCO invokes a GetValue() request then the LMS shall return an empty characterstring and set the error code to 403.- Data Model Element Value Not Initialized
REQ_60.4	If a SCO attempts to retrieve the **cmi.completion_threshold** and no completion threshold was defined in the Content Package Manifest (**<adlcp:completionThreshold>** element), then the LMS shall adhere to the GetValue API method requirements (Data Model Element Value Not Initialized).
REQ_61	The LMS shall implement the **cmi.credit** data model element.
REQ_61.1	The LMS shall implement the **cmi.credit** data model element as read-only.
REQ_61.2	The LMS shall implement the **cmi.credit** data model element as a state consisting of the following vocabulary tokens: • `credit` • `no-credit`
REQ_61.3	The LMS shall initialize the value of the **cmi.credit** data model element to the default value of `credit`.
REQ_62	The LMS shall implement the **cmi.entry** data model element.
REQ_62.1	The LMS shall implement the **cmi.entry** data model element as read-only.
REQ_62.2	The LMS shall implement the **cmi.entry** data model element as a state consisting of the following vocabulary tokens: • `ab-initio` • `resume`

REQ ID	Requirement
	• "" (empty characterstring)
REQ_62.3	The LMS shall initialize the **cmi.entry** data model element based on the following rules:
REQ_62.3.1	If this is the first learner session on a learner attempt, then the LMS shall set the **cmi.entry** to `ab-initio` prior to initial launch of the SCO. **ADL Note:** Upon a subsequent call to `GetValue(cmi.entry)`, the LMS shall return `ab-initio`.
REQ_62.3.2	If the learner attempt on the SCO is being resumed from a suspended learner session (**cmi.exit** is set to `suspend`), then the LMS shall initialize the **cmi.entry** value to `resume`. **ADL Note:** Upon a subsequent call to `GetValue(cmi.entry)`, the LMS shall return `resume`.
REQ_62.3.3	For all other conditions, the LMS shall set the **cmi.entry** to an empty characterstring ("").
REQ_63	The LMS shall implement the **cmi.exit** data model element.
REQ_63.1	The LMS shall implement the **cmi.exit** data model element as write-only.
REQ_63.2	The LMS shall implement the **cmi.exit** data model element as a state consisting of the following vocabulary tokens: • `time-out` • `suspend` • `logout` (**ADL Note:** This value is being deprecated and should not be used) • `normal` • `""` (empty characterstring)
REQ_63.3	The LMS shall initialize the value of the **cmi.exit** data model element to the default value of `""` (empty characterstring).
REQ_63.4	The LMS shall implement **cmi.exit** such that it has the following effect on sequencing behaviors:
REQ_63.4.1	If the SCO sets **cmi.exit** to `time-out`, the LMS shall process an "Exit All" navigation request when the SCO is taken away, instead of any pending (from the learner or LMS) navigation request.
REQ_63.4.2	If the SCO sets **cmi.exit** to `suspend`, the LMS sequencing implementation shall behave as if the Activity is Suspended value of the learning activity associated with the SCO is `true`.
REQ_63.4.3	If the SCO sets **cmi.exit** to `logout`, the LMS sequencing implementation shall process a "Exit All" navigation request when the SCO is taken away, instead of any pending (from the learner or LMS) navigation request. **ADL Note:** The value of `logout` is being deprecated and should not be used. If the value is used by a SCO, the LMS should adhere to this requirement.
REQ_63.5	If there are additional learner sessions within a learner attempt, the **cmi.exit** data model element's value shall be reset back to the default value of an empty characterstring ("") at the beginning of each additional learner session within the

REQ ID	Requirement
	learner attempt.
REQ_64	The LMS shall support at least the SPM of 250 interactions.
REQ_64.1	The LMS shall implement the **cmi.interactions._children** data model element.
REQ_64.1.1	The LMS shall implement the **cmi.interactions._children** data model element as read-only.
REQ_64.1.2	The LMS shall implement the **cmi.interactions._children** data model element as a characterstring.
REQ_64.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.interactions._children**, then the LMS shall return a comma-separated list of all **cmi.interactions** child data model elements • `id` • `type` • `objectives` • `timestamp` • `correct_responses` • `weighting` • `learner_response` • `result` • `latency` • `description` **ADL NOTE:** The order of these values is not significant.
REQ_64.2	The LMS shall implement the **cmi.interactions._count** data model element.
REQ_64.2.1	The LMS shall implement the **cmi.interactions._count** data model element as read-only.
REQ_64.2.2	The LMS shall implement the **cmi.interactions._count** data model element as a non-negative integer.
REQ_64.2.3	If a SCO invokes a GetValue() request to retrieve the **cmi.interactions._count**, then the LMS shall return the number of interactions currently stored by the LMS.
REQ_64.3	The LMS shall implement the **cmi.interactions.n.id** data model element.
REQ_64.3.1	The LMS shall implement the **cmi.interactions.n.id** data model element as a read/write.
REQ_64.3.2	The LMS shall implement the **cmi.interactions.n.id** data model element as a long_identifier_type with an SPM of 4000 characters.
REQ_64.3.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.id** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.3.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.id** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.

REQ ID	Requirement
REQ_64.3.5	If the SCO invokes a SetValue() request on the **cmi.interactions.n.id** data model element and the value of the identifier is not unique (i.e., another identifier exists in the collection of interactions at a different array position (n)), then the LMS shall process the request as it would normally. The interactions collection is defined as a bag of interaction records and duplicate entries (uniqueness is determined based on the **cmi.interactions.n.id** value) are permitted in the collection of interactions.
REQ_64.4	The LMS shall implement the **cmi.interactions.n.type** data model element.
REQ_64.4.1	The LMS shall implement the **cmi.interactions.n.type** data model element as read/write.
REQ_64.4.2	The LMS shall implement the **cmi.interactions.n.type** data model element as a state consisting of the following vocabulary tokens: • `true-false` • `choice` • `fill-in` • `long-fill-in` • `likert` • `matching` • `performance` • `sequencing` • `numeric` • `other`
REQ_64.4.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.type** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.4.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.type** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.5	The LMS shall support at least 10 (SPM) interaction's objectives.
REQ_64.5.1	The LMS shall implement the **cmi.interactions.n.objectives._count** data model element.
REQ_64.5.1.1	The LMS shall implement the **cmi.interactions.n.objectives._count** data model element as read-only.
REQ_64.5.1.2	The LMS shall implement the **cmi.interactions.n.objectives._count** data model element as a non-negative integer.
REQ_64.5.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.interactions.n.objectives._count**, then the LMS shall return the number of interaction objectives currently stored by the LMS.
REQ_64.5.2	The LMS shall implement the **cmi.interactions.n.objectives.m.id** data model element.

REQ ID	Requirement
REQ_64.5.2.1	The LMS shall implement **cmi.interactions.n.objectives.m.id** data model element as read/write.
REQ_64.5.2.2	The LMS shall implement the **cmi.interactions.n.objectives.m.id** data model element as a long_identifier_type with an SPM of 4000 characters.
REQ_64.5.2.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.objectives.m.id** data model element where the index (m) provided is a number that is greater than the current number of **cmi.interactions.n.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.5.2.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.objectives.m.id** data model element where the index (m) provided is a number that is greater than the current number of **cmi.interactions.n.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.5.2.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.objectives.m.id**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.5.2.6	If a SCO invokes a SetValue() request on the **cmi.interactions.n.objectives.m.id** where the value of the id is identical to an id that is already being persisted by the LMS for the given interaction, then the LMS shall set the error code to General Set Failure (351), return false and not persist the value being set.
REQ_64.6	The LMS shall implement the **cmi.interactions.n.timestamp** data model element.
REQ_64.6.1	The LMS shall implement the **cmi.interactions.n.timestamp** data model element as read/write.
REQ_64.6.2	The LMS shall implement the **cmi.interactions.n.timestamp** data model element as a time (second,10,0). This value shall be accurate to one second.
REQ_64.6.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.timestamp** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.6.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.timestamp** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.6.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.timestamp**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.7	The LMS shall implement the **cmi.interactions.n.correct_responses** data model elements. **ADL NOTE:** The SPM value for the number of correct_responses that the LMS shall support changes based on the **cmi.interactions.n.type** element.

REQ ID	Requirement
REQ_64.7.1	The LMS shall implement the **cmi.interactions.n.correct_responses._count** data model element.
REQ_64.7.1.1	The LMS shall implement the **cmi.interactions.n.correct_reponses._count** data model element as read-only.
REQ_64.7.1.2	The LMS shall implement the **cmi.interactions.n.correct_responses._count** data model element as a non-negative integer.
REQ_64.7.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.interactions.n.correct_responses._count**, then the LMS shall return the number of correct responses currently stored by the LMS.
REQ_64.7.2	The LMS shall implement the **cmi.interactions.n.correct_responses.m.pattern** data model element.
REQ_64.7.2.1	The LMS shall implement the **cmi.interactions.n.correct_responses.m.pattern** data model element as read/write.
REQ_64.7.2.2	The LMS shall implement the **cmi.interactions.n.correct_responses.m.pattern** data model element to adhere to the appropriate interaction type (**cmi.interactions.n.type**). This type changes based on the value of the **cmi.interactions.n.type** data model element. Refer to *Section 2.1.4 Run-Time Environment Data Model Data Type Conformance Requirements* for details on the datatype of correct_responses.
REQ_64.7.2.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.correct_responses.m.pattern** data model element where the index (m) provided is a number that is greater than the current number of **cmi.interactions.n.correct_responses** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.7.2.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.correct_responses.m.pattern** data model element where the index (m) provided is a number that is greater than the current number of **cmi.interactions.n.correct_responses** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.7.2.5	If the **cmi.interactions.n.id** and **cmi.interactions.n.type** have not been set prior to the request to set the **cmi.interactions.n.correct_responses.m.pattern**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.7.2.6	If the interaction type (**cmi.interactions.n.type**) is `true-false` then the LMS shall manage 1 and only 1 correct response pattern.
REQ_64.7.2.6.1	If a SCO attempts to set more than one correct response pattern then the LMS shall set the error code to 351 and return `false`.
REQ_64.7.2.7	If the interaction type (**cmi.interactions.n.type**) is `choice` then the LMS shall manage at least 10 correct response patterns (SPM requirement).
REQ_64.7.2.7.1	If the interaction type (**cmi.interactions.n.type**) is `choice` then correct response patterns shall be unique for a SCO.
REQ_64.7.2.7.1.1	If a correct response pattern for a choice interaction type, within the scope of a SCO, is found not to be unique with respect to the other correct response patterns managed by the LMS for the SCO, then the LMS shall set the error code to 351 and return `false`.

REQ ID	Requirement
REQ_64.7.2.8	If the interaction type (**cmi.interactions.n.type**) is `fill-in` then the LMS shall manage at least 5 correct response patterns (SPM requirement).
REQ_64.7.2.9	If the interaction type (**cmi.interactions.n.type**) is `long-fill-in` then the LMS shall manage at least 5 correct response patterns (SPM requirement).
REQ_64.7.2.10	If the interaction type (**cmi.interactions.n.type**) is `likert` then the LMS shall manage 1 and only 1 correct response pattern.
REQ_64.7.2.10.1	If a SCO attempts to set more than one correct response pattern then the LMS shall set the error code to 351 and return `false`.
REQ_64.7.2.11	If the interaction type (**cmi.interactions.n.type**) is `matching` then the LMS shall manage at least 5 correct response patterns (SPM requirement).
REQ_64.7.2.12	If the interaction type (**cmi.interactions.n.type**) is `performance` then the LMS shall manage at least 5 correct response patterns (SPM requirement).
REQ_64.7.2.13	If the interaction type (**cmi.interactions.n.type**) is `sequencing` then the LMS shall manage at least 5 correct response patterns (SPM requirement).
REQ_64.7.2.13.1	If the interaction type (**cmi.interactions.n.type**) is `sequencing` then correct response patterns shall be unique for a SCO.
REQ_64.7.2.13.1.1	If a correct response pattern for a sequencing interaction type, within the scope of a SCO, is found not to be unique with respect to the other correct response patterns managed by the LMS for the SCO, then the LMS shall set the error code to 351 and return `false`.
REQ_64.7.2.14	If the interaction type (**cmi.interactions.n.type**) is `numeric` then the LMS shall manage 1 and only 1 correct response pattern.
REQ_64.7.2.14.1	If a SCO attempts to set more than one correct response pattern then the LMS shall set the error code to 351 and return `false`.
REQ_64.7.2.15	If the interaction type (**cmi.interactions.n.type**) is `other` then the LMS shall manage 1 and only 1 correct response pattern.
REQ_64.7.2.15.1	If a SCO attempts to set more than one correct response pattern then the LMS shall set the error code to 351 and return `false`.
REQ_64.8	The LMS shall implement the **cmi.interactions.n.weighting** data model element.
REQ_64.8.1	The LMS shall implement the **cmi.interactions.n.weighting** data model element as read/write.
REQ_64.8.2	The LMS shall implement the **cmi.interactions.n.weighting** data model element as a real (10,7).
REQ_64.8.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.weighting** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.8.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.weighting** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.

REQ ID	Requirement
REQ_64.8.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.weighting**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.9	The LMS shall implement the **cmi.interactions.n.learner_response** data model element.
REQ_64.9.1	The LMS shall implement the **cmi.interactions.n.learner_response** data model element as read/write.
REQ_64.9.2	The LMS shall implement the **cmi.interactions.n.learner_response** data model element to adhere to the appropriate interaction type (**cmi.interactions.n.type**). This type changes based on the value of the **cmi.interactions.n.type** data model element. See the Data Model Datatype conformance requirements for details on the datatype of **cmi.interactions.n.learner_response**.
REQ_64.9.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.learner_response** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.9.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.learner_response** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.9.5	If the **cmi.interactions.n.id** and **cmi.interactions.n.type** have not been set prior to the request to set the **cmi.interactions.n.learner_response**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.10	The LMS shall implement the **cmi.interactions.n.result** data model element.
REQ_64.10.1	The LMS shall implement the **cmi.interactions.n.result** data model element as read/write.
REQ_64.10.2	The LMS shall implement the **cmi.interactions.n.result** data model element as a state consisting of the following vocabulary tokens: • `correct` • `incorrect` • `unanticipated` • `neutral` • `real(10,7)`
REQ_64.10.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.result** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").

REQ ID	Requirement
REQ_64.10.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.result** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.10.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.result**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.11	The LMS shall implement the **cmi.interactions.n.latency** data model element.
REQ_64.11.1	The LMS shall implement the **cmi.interactions.n.latency** data model element as read/write.
REQ_64.11.2	The LMS shall implement the **cmi.interactions.n.latency** data model element as a timeinterval (second,10,2).
REQ_64.11.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.latency** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.11.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.latency** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_64.11.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.latency**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_64.12	The LMS shall implement the **cmi.interactions.n.description** data model element.
REQ_64.12.1	The LMS shall implement the **cmi.interactions.n.description** data model element as a read/write.
REQ_64.12.2	The LMS shall implement the **cmi.interactions.n.description** data model element as a localized_string_type with an SPM of 250 characters.
REQ_64.12.3	If the SCO invokes a GetValue() request on the **cmi.interactions.n.description** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_64.12.4	If the SCO invokes a SetValue() request on the **cmi.interactions.n.description** data model element where the index (n) provided is a number that is greater than the current number of **cmi.interactions** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.

REQ ID	Requirement
REQ_64.12.5	If the **cmi.interactions.n.id** has not been set prior to the request to set the **cmi.interactions.n.description**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_65	The LMS shall implement the **cmi.launch_data** data model element.
REQ_65.1	The LMS shall implement the **cmi.launch_data** data model element as read-only.
REQ_65.2	The LMS shall implement the **cmi.launch_data** data model element as a characterstring with an SPM of 4000 characters.
REQ_65.3	The LMS shall initialize this data model element using the SCORM 2004 3rd Edition Content Packaging Extensions Version 1.0 namespace element **<adlcp:dataFromLMS>**. If an **<adlcp:dataFromLMS>** element does not exist as a child element of the **<imscp:item>** element (associated with the SCO resource), then the element shall remain uninitialized.
REQ_66	The LMS shall implement the **cmi.learner_id** data model element.
REQ_66.1	The LMS shall implement the **cmi.learner_id** data model element as read-only.
REQ_66.2	The LMS shall implement the **cmi.learner_id** data model element as a long_identifer_type with an SPM of 4000 characters.
REQ_66.3	The LMS shall be responsible for initializing the **cmi.learner_id**. **ADL NOTE:** How this is done is currently outside the scope of SCORM (e.g., this is typically handled via a learner registration system within the LMS).
REQ_67	The LMS shall implement the **cmi.learner_name** data model element.
REQ_67.1	The LMS shall implement the **cmi.learner_name** data model element as read-only.
REQ_67.2	The LMS shall implement the **cmi.learner_name** data model element as a localized_string_type with an SPM of 250 characters.
REQ_67.3	The LMS shall be responsible for initializing the **cmi.learner_name**. **ADL NOTE:** How this is done is currently outside the scope of SCORM (e.g., this is typically handled via a learner registration system within the LMS).
REQ_68	The LMS shall implement the **cmi.learner_preference** data model elements as follows:
REQ_68.1	The LMS shall implement the **cmi.learner_preference._children** data model element.
REQ_68.1.1	The LMS shall implement the **cmi.learner_preference._children** data model element as read-only.
REQ_68.1.2	The LMS shall implement the **cmi.learner_preference._children** data model element as a characterstring.
REQ_68.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.learner_preference._children**, then the LMS shall return a comma-separated list of all learner preference child data model elements: • `audio_level` • `language`

REQ ID	Requirement
	• `delivery_speed` • `audio_captioning` **ADL NOTE:** The order of these values is not significant.
REQ_68.2	The LMS shall implement the **cmi.learner_preference.language** data model element.
REQ_68.2.1	The LMS shall implement the **cmi.learner_preference.language** data model element as read/write.
REQ_68.2.2	The LMS shall implement the **cmi.learner_preference.language** data model element as a language_type with an SPM of 250 characters or an empty characterstring ("").
REQ_68.2.3	If the LMS does not provide a mechanism for initializing the learner preferences, then the default value for the **cmi.learner_preference.language** should be an empty characterstring "". This value should be used to initialize the data model element.
REQ_68.3	The LMS shall implement the **cmi.learner_preference.delivery_speed** data model element.
REQ_68.3.1	The LMS shall implement the **cmi.learner_preference.delivery_speed** data model element as read/write.
REQ_68.3.2	The LMS shall implement the **cmi.learner_preference.delivery_speed** data model element as real (10,7). The data model element shall support a range between (0.0 .. *).
REQ_68.3.3	If an LMS does not provide a mechanism for initializing the learner preferences, then the default value for the **cmi.learner_preference.delivery_speed** shall be `1`. This value should be used to initialize the data model element.
REQ_68.4	The LMS shall implement the **cmi.learner_preference.audio_captioning** data model element.
REQ_68.4.1	The LMS shall implement the **cmi.learner_preference.audio_captioning** data model element as read/write.
REQ_68.4.2	The LMS shall implement the **cmi.learner_preference.audio_captioning** data model element as a state consisting of the following vocabulary tokens: • `-1` • `0` • `1`
REQ_68.4.3	If an LMS does not provide a mechanism for initializing the learner preferences, then the default value for the **cmi.learner_preference.audio_captioning** shall be `0`. This value should be used to initialize the data model element.
REQ_68.5	The LMS shall implement the **cmi.learner_preference.audio_level** data model element.
REQ_68.5.1	The LMS shall implement the **cmi.learner_preference.audio_level** data model element as read/write.
REQ_68.5.2	The LMS shall implement the **cmi.learner_preference.audio_level** data model element as real (10,7). The data model element shall support a value that is greater than or equal to 0.
REQ_68.5.3	If an LMS does not provide a mechanism for initializing the learner preferences,

REQ ID	Requirement
	then the default value for the **cmi.learner_preference.audio_level** shall be `1`. This value should be used to initialize the data model element.
REQ_69	The LMS shall implement the **cmi.location** data model element.
REQ_69.1	The LMS shall implement the **cmi.location** data model element as read/write.
REQ_69.2	The LMS shall implement the **cmi.location** data model element as a characterstring with an SPM of 1000 characters.
REQ_70	The LMS shall implement the **cmi.max_time_allowed** data model element.
REQ_70.1	The LMS shall implement the **cmi.max_time_allowed** data model element as read-only.
REQ_70.2	The LMS shall implement **cmi.max_time_allowed** as a timeinterval (second,10,2).
REQ_70.3	The LMS shall initialize this data model element using the IMS Simple Sequencing namespace element **<imsss:attemptAbsoluteDurationLimit>**. If an **<imsss:attemptAbsoluteDurationLimit>** element does not exist as a child element of the **<imscp:item>** element (associated with the SCO resource), then the element shall remain uninitialized. **NOTE:** In this case uninitialized is defined to indicate that no value should be assigned to cmi.max_time_allowed. In this case, if a SCO invokes a GetValue() request then the LMS shall return an empty characterstring and set the error code to 403.- Data Model Element Value Not Initialized
REQ_71	The LMS shall implement the **cmi.mode** data model element.
REQ_71.1	The LMS shall implement the **cmi.mode** data model element as read-only.
REQ_71.2	The LMS shall implement the **cmi.mode** data model element as a state consisting of the following vocabulary tokens: • `browse` • `normal` • `review`
REQ_71.3	If a mechanism is not in place to support different modes, then the LMS shall initialize the **cmi.mode** data model element to the default value, `normal`.
REQ_71.4	If the **cmi.mode** value is `browse` or `review`, the LMS shall treat any data sent by the SCO as informative (in order to make sequencing decisions). Whether or not an LMS persists any of the data sent by the SCO, while in a mode of review or browse, is outside the scope of the SCORM.
REQ_72	The LMS shall support at least the SPM of 100 objectives.
REQ_72.1	The LMS shall implement the **cmi.objectives._children** data model element.
REQ_72.1.1	The LMS shall implement the **cmi.objectives._children** data model element as read-only.
REQ_72.1.2	The LMS shall implement the **cmi.objectives._children** data model element as a characterstring.

REQ ID	Requirement
REQ_72.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.objectives._children**, then the LMS shall return a comma-separated list of all objectives child data model elements: • `id` • `score` • `success_status` • `completion_status` • `progress_measure` • `description` **ADL NOTE:** The order of these values is not significant.
REQ_72.2	The LMS shall implement the **cmi.objectives._count** data model element.
REQ_72.2.1	The LMS shall implement the **cmi.objectives._count** data model element as read-only.
REQ_72.2.2	The LMS shall implement the **cmi.objectives._count** data model element as a non-negative integer.
REQ_72.2.3	If a SCO invokes a GetValue() request to retrieve the **cmi.objectives._count**, then the LMS shall return the number of objectives currently stored by the LMS.
REQ_72.3	The LMS shall implement the **cmi.objectives.n.id** data model element.
REQ_72.3.1	The LMS shall implement the **cmi.objectives.n.id** data model element as read/write.
REQ_72.3.2	The LMS shall implement the **cmi.objectives.n.id** data model element as a long_identifier_type with an SPM of 4000 characters.
REQ_72.3.3	For each objective defined (**<imsss:primaryObjective>** or **<imsss:objective>**) that includes an objectiveID attribute, the LMS is responsible for adhering to the following initialization requirements:
REQ_72.3.3.1	The **objectiveID** attribute shall be used to initialize the **cmi.objectives.n.id** value.
REQ_72.3.3.2	If a Read Objective Normalized Measure map is defined (in a **<imsss:mapInfo>** element) and the sequencing implementation is maintaining a valid Objective Normalize Measure (Objective Measure Status is `true`), then the Objective Normalized Measure shall be used to initialize the **cmi.objectives.n.score.scaled**.
REQ_72.3.3.3	If a Read Objective Satisfied Status map is defined (in a **<imsss:mapInfo>** element) and the sequencing implementation is maintaining a valid Objective Satisfied Status (Objective Progress Status is `true`), then the Objective Satisfied Status shall be used to initialize the **cmi.objectives.n.success_status** as defined by the following requirements.
REQ_72.3.3.3.1	If the Objective Satisfied Status is `true`, then the **cmi.objectives.n.success_status** shall be initialized to `passed`.
REQ_72.3.3.3.2	If the Objective Satisfied Status is `false`, then the **cmi.objectives.n.success_status** shall be initialized to `failed`.

REQ ID	Requirement
REQ_72.3.4	If the SCO invokes a GetValue() request on the **cmi.objectives.n.id** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.3.5	If the SCO invokes a SetValue() request on the **cmi.objectives.n.id** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.3.6	If the SCO invokes a SetValue() request on the **cmi.objectives.n.id** data model element and the value of the identifier is not unique (i.e., another identifier exists in the collection of objectives at a different array position (n)), then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.4	The LMS shall implement the **cmi.objectives.n.score** data model elements.
REQ_72.4.1	The LMS shall implement the **cmi.objectives.n.score._children** data model element.
REQ_72.4.1.1	The LMS shall implement the **cmi.objectives.n.score._children** data model element as read-only.
REQ_72.4.1.2	The LMS shall implement the **cmi.objectives.n.score._children** data model element as a characterstring.
REQ_72.4.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.objectives.n.score._children**, then the LMS shall return a comma-separated list of all of the scores child data model elements: • `scaled` • `raw` • `min` • `max` **ADL NOTE:** The order of these values is not significant.
REQ_72.4.2	The LMS shall implement the **cmi.objectives.n.score.scaled** data model element.
REQ_72.4.2.1	The LMS shall implement the **cmi.objectives.n.score.scaled** data model element as read/write.
REQ_72.4.2.2	The LMS shall implement the **cmi.objectives.n.score.scaled** data model element as a real (10,7). The value shall be in the range of -1.0 to 1.0, inclusive.
REQ_72.4.2.3	The LMS shall implement **cmi.objectives.n.score.scaled** such that it has the following effect on sequencing behaviors:
REQ_72.4.2.3.1	If the SCO does not set **cmi.objectives.n.score.scaled** for an objective of the SCO, then the LMS' sequencing implementation shall behave as if the Objective Measure Status for the associated objective (based on objective IDs) of the learning activity associated with the SCO is `false`.

REQ ID	Requirement
REQ_72.4.2.3.2	If the SCO sets **cmi.objectives.n.score.scaled** for an objective of the SCO, then the LMS' sequencing implementation shall behave as if the Objective Measure Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `true`, and the Objective Normalized Measure for the objective (based on objective IDs) of the learning activity associated with the SCO is equal to the value of **cmi.objectives.n.score.scaled**.
REQ_72.4.2.4	If the SCO invokes a GetValue() request on the **cmi.objectives.n.score.scaled** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.4.2.5	If the SCO invokes a SetValue() request on the **cmi.objectives.n.score.scaled** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.4.2.6	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.score.scaled** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.4.3	The LMS shall implement the **cmi.objectives.n.score.raw** data model element.
REQ_72.4.3.1	The LMS shall implement the **cmi.objectives.n.score.raw** data model element as read/write.
REQ_72.4.3.2	The LMS shall implement **cmi.objectives.n.score.raw** as a real (10,7).
REQ_72.4.3.3	If the SCO invokes a GetValue() request on the **cmi.objectives.n.score.raw** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.4.3.4	If the SCO invokes a SetValue() request on the **cmi.objectives.n.score.raw** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.4.3.5	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.score.raw** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.4.4	The LMS shall implement the **cmi.objectives.n.score.min** data model element.
REQ_72.4.4.1	The LMS shall implement the **cmi.objectives.n.score.min** data model element as read/write.
REQ_72.4.4.2	The LMS shall implement **cmi.objectives.n.score.min** as a real (10,7).

REQ ID	Requirement
REQ_72.4.4.3	If the SCO invokes a GetValue() request on the **cmi.objectives.n.score.min** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.4.4.4	If the SCO invokes a SetValue() request on the **cmi.objectives.n.score.min** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.4.4.5	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.score.min** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.4.5	The LMS shall implement the **cmi.objectives.n.score.max** data model element.
REQ_72.4.5.1	The LMS shall implement the **cmi.objectives.n.score.max** data model element as read/write.
REQ_72.4.5.2	The LMS shall implement **cmi.objectives.n.score.max** as a real (10,7).
REQ_72.4.5.3	If the SCO invokes a GetValue() request on the **cmi.objectives.n.score.max** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.4.5.4	If the SCO invokes a SetValue() request on the **cmi.objectives.n.score.max** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.4.5.5	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.score.max** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.5	The LMS shall implement the **cmi.objectives.n.success_status** data model element.
REQ_72.5.1	The LMS shall implement the **cmi.objectives.n.success_status** data model element as read/write.
REQ_72.5.2	The LMS shall implement the **cmi.objectives.n.success_status** data model element as a state consisting of the following vocabulary tokens: • `passed` • `failed` • `unknown`
REQ_72.5.3	The LMS shall initialize the value of the **cmi.objectives.n.success_status** data model element to the default value of `unknown`.

REQ ID	Requirement
REQ_72.5.4	The LMS shall implement **cmi.objectives.n.success_status** such that it has the following effect on sequencing behaviors:
REQ_72.5.4.1	If the SCO sets **cmi.objectives.n.success_status** for an objective of the SCO to `unknown`, the LMS' sequencing implementation shall behave as if the Objective Progress Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `false`.
REQ_72.5.4.2	If the SCO sets **cmi.objectives.n.success_status** for an objective of the SCO to `passed`, the LMS' sequencing implementation shall behave as if the Objective Progress Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `true`, and the Objective Satisfied Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `true`.
REQ_72.5.4.3	If the SCO sets **cmi.objectives.n.success_status** for an objective of the SCO to `failed`, the LMS' sequencing implementation shall behave as if the Objective Progress Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `true`, and the Objective Satisfied Status for the objective (based on objective IDs) of the learning activity associated with the SCO is `false`.
REQ_72.5.5	If the SCO invokes a GetValue() request on the **cmi.objectives.n.success_status** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.5.6	If the SCO invokes a SetValue() request on the **cmi.objectives.n.success_status** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.5.7	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.success_status** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.6	The LMS shall implement the **cmi.objectives.n.completion_status** data model element.
REQ_72.6.1	The LMS shall implement the **cmi.objectives.n.completion_status** data model element as a read/write.
REQ_72.6.2	The LMS shall implement the **cmi.objectives.n.completion_status** data model element as a state consisting of the following vocabulary tokens: • `completed` • `incomplete` • `not attempted` • `unknown`
REQ_72.6.3	If the SCO invokes a GetValue() request on the **cmi.objectives.n.completion_status** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").

REQ ID	Requirement
REQ_72.6.4	If the SCO invokes a SetValue() request on the **cmi.objectives.n.completion_status** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.6.5	Since the **cmi.objectives.n.id** is required to be set first prior to any other objective information, if the SCO attempts to set **cmi.objectives.n.completion_status** (prior to setting the identifier) then the LMS shall set the error code to Data Model Dependency Not Established (408) and return `false`. The LMS shall not alter the state of the data model element based on the request.
REQ_72.7	The LMS shall implement the **cmi.objectives.n.description** data model element.
REQ_72.7.1	The LMS shall implement the **cmi.objectives.n.description** data mode element as read/write.
REQ_72.7.2	The LMS shall implement the **cmi.objectives.n.description** data model element as a localized_string_type with an SPM of 250 characters.
REQ_72.7.3	If the SCO invokes a GetValue() request on the **cmi.objectives.n.description** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Get Failure (301) and return an empty characterstring ("").
REQ_72.7.4	If the SCO invokes a SetValue() request on the **cmi.objectives.n.description** data model element where the index (n) provided is a number that is greater than the current number of **cmi.objectives** being stored, then the LMS shall set the error code to General Set Failure (351) and return `false`.
REQ_72.7.5	If the **cmi.objectives.n.id** has not been set prior to the request to set the **cmi.objectives.n.description**, then the LMS shall set the error code to Data Model Dependency Not Established (408), return `false` and not change the current state of the data model element.
REQ_72.8	The LMS shall implement the **cmi.objectives.n.progress_measure** data model element.
REQ_72.8.1	The LMS shall implement the **cmi.objectives.n.progress_measure** data model element as read/write.
REQ_72.8.2	The LMS shall implement the **cmi.objectives.n.progress_measure** data model element as a real (10,7). The value shall be in the range of 0.0 to 1.0, inclusive.
REQ_73	The LMS shall implement the **cmi.progress_measure** data model element.
REQ_73.1	The LMS shall implement the **cmi.progress_measure** data model element as read/write.
REQ_73.2	The LMS shall implement the **cmi.progress_measure** data model element as a real (10,7). The value shall be in the range of 0.0 to 1.0, inclusive.
REQ_74	The LMS shall implement the **cmi.scaled_passing_score** data model element.
REQ_74.1	The LMS shall implement the **cmi.scaled_passing_score** data model element as read-only.
REQ_74.2	The LMS shall implement the **cmi.scaled_passing_score** data model element as a real (10,7). The value shall be in the range of -1.0 to 1.0, inclusive.

REQ ID	Requirement
REQ_74.3	The LMS is responsible for initializing the **cmi.scaled_passing_score** data model element using the IMS Simple Sequencing namespace element **<imsss:minNormalizedMeasure>** associated with an **<imsss:primaryObjective>** element for the **<imscp:item>** element that references a SCO resource as defined by the following requirements.
REQ_74.3.1	If the IMS Simple Sequencing namespace attribute **imsss:satisfiedByMeasure** associated with the **<imsss:primaryObjective>** element for the **<imscp:item>** element that references the SCO is equal to true, then the value provided by the **<imsss:minNormalizedMeasure>** element associated with the **<imsss:primaryObjective>** element for the **<imscp:item>** element that references the SCO resource shall be used to initialize the **cmi.scaled_passing_score** data model element.
REQ_74.3.2	If the IMS Simple Sequencing namespace attribute **imsss:satisfiedByMeasure** associated with the **<imsss:primaryObjective>** element for the **<imscp:item>** element that references the SCO is equal to true and no value is provided for the **<imsss:minNormalizedMeasure>** element associated with the **<imsss:primaryObjective>** element for the **<imscp:item>** element that references the SCO resource, then the LMS shall initialize the **cmi.scaled_passing_score** to 1.0.
REQ_74.3.3	If the IMS Simple Sequencing namespace attribute **imsss:satisfiedByMeasure** associated with the **<imsss:primaryObjective>** element for the **<imscp:item>** element that references the SCO is equal to false, then the LMS shall not make any assumptions of a scaled passing score (i.e., **cmi.scaled_passing_score** data model element**)**.
REQ_75	The LMS shall implement the **cmi.score** data model elements.
REQ_75.1	The LMS shall implement the **cmi.score._children** data model element.
REQ_75.1.1	The LMS shall implement the **cmi.score._children** data model element as read-only.
REQ_75.1.2	The LMS shall implement the **cmi.score._children** data model element as a characterstring.
REQ_75.1.3	If a SCO invokes a GetValue() request to retrieve the **cmi.score._children**, then the LMS shall return a comma-separated list of all of the score child data model elements: • `scaled` • `raw` • `min` • `max` **ADL NOTE:** The order of these values is not significant.
REQ_75.2	The LMS shall implement the **cmi.score.scaled** data model element.
REQ_75.2.1	The LMS shall implement the **cmi.score.scaled** data model element as read/write.
REQ_75.2.2	The LMS shall implement the **cmi.score.scaled** data model element as a real (10,7) The value shall be in the range of -1.0 to 1.0, inclusive.
REQ_75.2.3	The LMS shall implement **cmi.score.scaled** such that it has the following effect on sequencing behaviors:

REQ ID	Requirement
REQ_75.2.3.1	If the SCO does not set **cmi.score.scaled**, then the LMS sequencing implementation shall behave as if the Objective Measure Status for the primary objective of the learning activity associated with the SCO is `false`.
REQ_75.2.3.2	If the SCO sets **cmi.score.scaled**, the LMS sequencing implementation shall behave as if the Objective Measure Status for the primary objective of the learning activity associated with the SCO is `true`, and the Objective Normalized Measure for the primary objective of the learning activity associated with the SCO is equal the value of **cmi.score.scaled**.
REQ_75.3	The LMS shall implement the **cmi.score.raw** data model element.
REQ_75.3.1	The LMS shall implement the **cmi.score.raw** data model element as read/write.
REQ_75.3.2	The LMS shall implement **cmi.score.raw** as a real (10,7).
REQ_75.4	The LMS shall implement the **cmi.score.max** data model element.
REQ_75.4.1	The LMS shall implement the **cmi.score.max** data model element as read/write.
REQ_75.4.2	The LMS shall implement **cmi.score.max** as a real (10,7).
REQ_75.5	The LMS shall implement the **cmi.score.min** data model element.
REQ_75.5.1	The LMS shall implement the **cmi.score.min** data model element as read/write.
REQ_75.5.2	The LMS shall implement **cmi.score.min** as a real (10,7).
REQ_76	The LMS shall implement the **cmi.session_time** data model element.
REQ_76.1	The LMS shall implement the **cmi.session_time** data model element as write-only.
REQ_76.2	The LMS shall implement the **cmi.session_time** data model element a timeinterval (seconds, 10, 2).
REQ_76.3	Since a SCO is not required to set a value for this data model element (not required to keep track of the session time), an LMS shall keep track of session time from the time the LMS launches the SCO. If the SCO reports a different session time, then the LMS shall use the session time as reported by the SCO instead of the session time as measured by the LMS.
REQ_76.4	When the SCO issues the **Terminate("")** or the user navigates away, the LMS shall take the last **cmi.session_time** that the SCO set (if there was a set) and accumulate this time to the **cmi.total_time**.
REQ_76.5	If there are additional learner sessions within a learner attempt, the cmi.session_time shall be reset at the beginning of each additional learner session within the learner attempt.
REQ_77	The LMS shall implement the **cmi.success_status** data model element.
REQ_77.1	The LMS shall implement the **cmi.success_status** data model element as read/write.
REQ_77.2	The LMS shall implement the **cmi.success_status** data model element as a state consisting of the following vocabulary tokens: • `passed` • `failed` • `unknown`

REQ ID	Requirement
REQ_77.3	The LMS shall initialize the value of the **cmi.success_status** data model element to the default value of `unknown`.
REQ_77.4	The LMS shall implement **cmi.success_status** such that it has the following effect on sequencing behaviors:
REQ_77.4.1	If the SCO or LMS sets **cmi.success_status**, of the SCO to `unknown`, then the LMS' sequencing implementation shall behave as if the Objective Progress Status for the primary objective of the learning activity associated with the SCO is `false`.
REQ_77.4.2	If the SCO or LMS sets **cmi.success_status**, of the SCO to `passed`, then the LMS' sequencing implementation shall behave as if the Objective Progress Status for the primary objective of the learning activity associated with the SCO is `true`, and the Objective Satisfied Status for the primary objective of the learning activity associated with the SCO is `true`.
REQ_77.4.3	If the SCO or LMS sets **cmi.success_status**, of the SCO to `failed`, then the LMS' sequencing implementation shall behave as if the Objective Progress Status for the primary objective of the learning activity associated with the SCO is `true`, and the Objective Satisfied Status for the primary objective of the learning activity associated with the SCO is `false`.
REQ_77.5	The LMS shall evaluate the value of the **cmi.success_status** data model element and return the result in the response to a `GetValue()` request according to the following requirements:
REQ_77.5.1	If a **cmi.scaled_passing_score** is defined for the SCO and the **cmi.score.scaled** data model element's value is set by the SCO and the value is greater than or equal to the **cmi.scaled_passing_score**, then the LMS shall evaluate and return the value of `passed`.
REQ_77.5.2	If a **cmi.scaled_passing_score** is defined for the SCO and the **cmi.score.scaled** data model element's value is set by the SCO and the value is less than the **cmi.scaled_passing_score**, then the LMS shall evaluate and return the value of `failed`.
REQ_77.5.3	If the **cmi.scaled_passing_score** has been defined and the SCO does not set a **cmi.score.scaled**, the LMS shall evaluate and return the value of `unknown`.
REQ_77.5.4	If no **cmi.scaled_passing_score** has been defined for the SCO, then the LMS shall rely on the value set for the **cmi.success_status** data model element by the SCO and return that value. If no value was set by the SCO for the **cmi.success_status** data model element then the LMS shall return `unknown`.
REQ_78	The LMS shall implement the **cmi.suspend_data** data model element.
REQ_78.1	The LMS shall implement the **cmi.suspend_data** data model element as read/write.
REQ_78.2	The LMS shall implement the **cmi.suspend_data** data model element as a characterstring with an SPM of 64000 characters.
REQ_79	The LMS shall implement the **cmi.time_limit_action** data model element.
REQ_79.1	The LMS shall implement the **cmi.time_limit_action** data model element as read-only.
REQ_79.2	The LMS shall implement the **cmi.time_limit_action** data model element as a state consisting of the following vocabulary tokens: • `exit,message`

REQ ID	Requirement
	• `continue,message` • `exit,no message` • `continue,no message`
REQ_79.3	The LMS shall initialize this data model element using the SCORM 2004 3rd Edition Content Packaging Extensions Version 1.0 namespace element **<adlcp:timeLimitAction>**. If an **<adlcp:timeLimitAction>** element does not exist as a child element of the **<imscp:item>** element (associated with the SCO resource), then the element shall be initialized to the default value of `continue,no message`.
REQ_80	The LMS shall implement the **cmi.total_time** data model element.
REQ_80.1	The LMS shall implement the **cmi.total_time** data model element as read-only.
REQ_80.2	The LMS shall implement the **cmi.total_time** data model element as a timeinterval (second, 10, 2).
REQ_80.3	The value of the **cmi.total_time** shall not be updated by the LMS while a learner session is in progress.
REQ_80.4	The default value for the **cmi.total_time** shall be any value represented in the defined data type format, that evaluates to an interval of time of zero.
REQ_90	The LMS shall adhere to the following general data model requirements:
REQ_90.1	The LMS shall implement all of the SCORM Run-Time Environment Data Model Elements.
REQ_90.2	All data model element names shall be bound to an ECMAScript characterstring using a dot-notation.
REQ_90.3	All arrays shall be implemented with a starting index of 0 (zero-based arrays).

2.1.4. Run-Time Environment Data Model Data Type Conformance Requirements

This section defines the conformance requirements of the datatypes for the SCORM Run-Time Environment Data Model elements. Table 2.1.4a outlines the requirements that implementations shall adhere to when processing data model elements.

Table 2.1.4a: Run-Time Environment Data Model Data Type Conformance Requirements

REQ ID	Requirement
REQ_81	A characterstring shall be a string of characters that are defined in ISO 10646. ISO 10646 is equivalent to the Unicode Standard.
REQ_82	A localized_string_type shall be a characterstring that has an indicator of the language of the characterstring.
REQ_82.1	The format of the characterstring shall have the following syntax: • `{lang=<language_type>}<actual characterstring>`
REQ_82.1.1	The {lang=<language_type>} reserved delimiter shall represent the language of the characterstring (i.e., <actual characterstring>).
REQ_82.1.2	The {lang=<language_type>} reserved delimiter shall be optionally used in a localized_string_type.
REQ_82.1.3	The default language identifier, if no {lang=<language_type>} is specified, shall be `en` (English).
REQ_83	The <language_type> shall be used to represent a language.
REQ_83.1	The format of a <language_type> shall be: <language_type> ::= langcode ["-" subcode]* where the langcode is either: • 2-letter codes as defined by ISO 639-1 • 3-letter codes as defined by ISO 639-2 • The value `i` is reserved and used as a prefix for registrations defined by Internet Assigned Numbers Authority (IANA) • The value `x` is reserved and used as a prefix for private use and the subcode is: • 2-letter subcodes are ISO 3166-1 alpha –2 country codes • subcodes of from 3 to 8 letters are registered with IANA ISO 639-2 specifies two code sets (bibliographic and terminology) for the language code. Either code set may be used. The langcode and subcode is case insensitive. SCORM recommends that the language_type be represented with the following format: • langcode is normally given in lower case, and • subcodes (if any) are normally in upper case. The langcode and subcodes shall have lengths ranging between a minimum of 1 character to a maximum of 8 characters. **ADL NOTE:** Refer to Appendix B for a listing of required language codes defined by ISO 639-1 and ISO 639-2, and IANA.

REQ ID	Requirement
REQ_83.2	The <language_type> shall be implemented with an SPM of 250 characters
REQ_84	The long_identifier_type shall be a characterstring that conforms to the syntax defined for Universal Resource Identifiers (URIs). The URI shall be conformant to RFC 3986.
REQ_84.1	This label or identifier shall be unique within the context of the SCO. **ADL NOTE:** Therefore, an empty characterstring is not a valid long_identifier_type.
REQ_84.2	The SCORM recommends that the URI be a Universal Resource Name (URN), if a URN is used the URN shall have the following syntax: <URN> ::= "urn:"<NID>":"<NSS> . The phrases in quotes are a required part of the URN. The <NID> is the namespace identifier and <NSS> is the namespace specific string. The URN shall be conformant to RFC 2141.
REQ_84.3	The long_identifier_type's value shall be implemented with an SPM of 4000 characters
REQ_85	The short_identifier_type shall be a characterstring that conforms to the syntax defined for Universal Resource Identifiers (URI). The URI shall be conformant to RFC 3986.
REQ_85.1	This label or identifier shall be unique within the context of the SCO. **ADL NOTE:** Therefore, an empty characterstring is not a valid short_identifier_type.
REQ_85.2	The short_identifier_type's value shall be implemented with an SPM of 250 characters
REQ_86	The integer datatype shall be a member of the set of positive whole numbers (i.e., 1,2,3, etc.), negative whole numbers (i.e., -1, -2, -3, etc.) and zero (0).
REQ_87	The real(10,7) datatype shall be a real number with a precision of seven significant digits.
REQ_88	The time (second,10,0) data type represents a point in time. This data type shall have a required precision of 1 second and an optional precision of 0.01 seconds

REQ ID	Requirement
REQ_88.1	The format of the characterstring shall be as follows: YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]]] where· • YYYY: A four-digit year (1970 <= YYYY<=2038) • MM: A two-digit month (01 through 12 where 01=January) • DD: A two-digit day of month (01 through 31, depending on the value of month and year) • hh: Two-digits of hour (00 through 23) • mm: Two-digits of minute (00 through 59) • ss: Two-digits of second (00 through 59) • s: 1 or more digits representing a decimal fraction of a second). If fractions of a second are used, SCORM further restricts the string to a maximum of 2 digits (e.g., 34.45 - valid, 34.45454545 - not valid) • TZD: Time zone designator (Z for UTC or +hh:mm or -hh:mm). o The hh and mm shall adhere to the requirements defined above for hh and mm. o If the difference between local time and UTC is required then the time can be expressed in hours and minutes (i.e., 03:10) or if the time difference is exactly an integral number of hours, then hours only (03). o The leading zero is required for hours less than 10. o If no TZD designator is provided then the default time zone shall be interpreted as "local time". o If TZD is defined, then hh:mm:ss.s shall also be defined, although they may be all zeros. At least the four-digit year must be present. If additional parts of the time are included, the character literals: • - • T • : • . are part of the character lexical representation.
REQ_89	The timeinterval (second,10,2) denotes that the value for the data model element timeinterval is a number expressed as a real datatype with a value that is accurate to one hundredths of a second.
REQ_89.1	The format of the characterstring shall be as follows: P[yY][mM][dD][T[hH][nM][s[.s]S]] where: • y: The number of years (integer, >= 0, not restricted) • m: The number of months (integer, >=0, not restricted) • d: The number of days (integer, >=0, not restricted) • h: The number of hours (integer, >=0, not restricted) • n: The number of minutes (integer, >=0, not restricted) • s: The number of seconds or fraction of seconds (real or integer, >=0, not restricted). If fractions of a second are used, SCORM further restricts the

REQ ID	Requirement
	string to a maximum of 2 digits (e.g., 34.45 - valid, 34.45454545 - not valid) • Zero-padding of the values shall be supported The character literals designators • P • Y • M • D • T • H • M • S shall appear if the corresponding non-zero value is present. If the data model element contains a value, then the designator P shall be present. If the value of years, months, days, hours, minutes or seconds is zero, then the value and corresponding designation may be omitted, but at least one designator and value shall be present in addition to the designator P. The designator T shall be omitted if all of the time components (hours, minutes, seconds) are not used. A zero value may be used with any of the time components (e.g., PT0S).
REQ_91	The format for the characterstring representation of **cmi.interactions.n.correct_responses.m.pattern** and **cmi.interactions.n.learner_reponse** change based on the **cmi.interactions.n.type** data model element's value. The following characterstring formats shall be enforced on the **cmi.interactions.n.correct_responses.m.pattern** and **cmi.interactions.n.learner_response** elements and the type shall be determined by the **cmi.interactions.n.type** data model element.
REQ_91.1	The value of the characterstring for the **cmi.interactions.n.correct_responses.m.pattern** shall adhere to the following requirements (dependent on the **cmi.interactions.n.type**).
REQ_91.1.1	If the **cmi.interactions.n.type** is `true-false`, then the correct response pattern shall be a characterstring containing one of the following reserved tokens: `true` and `false`.
REQ_91.1.2	If the **cmi.interactions.n.type** is `choice`, then the correct response pattern shall be a characterstring value that represents a set of short_identifier_types, where each element of the set is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 36 (the required SPM) short_identifier_types. The characterstring shall be represented in the following format: • <short_identifier_type>[,]<short_identifier_type> The following requirements shall be adhered to when building the characterstring:
REQ_91.1.2.1	If the correct response pattern is to represent that there is no correct choice then the set shall be represented as an empty characterstring (“”). **ADL NOTE:** The empty characterstring implies an empty set; therefore there are no short_identifier_types in the set.

REQ ID	Requirement
REQ_91.1.2.2	If the correct response pattern has multiple short_identifier_types, all of which are required, then each short_identifier_type shall be separated by the special reserved token `[,]`.
REQ_91.1.2.2.1	Each short_identifier_type shall be unique within the correct response pattern.
REQ_91.1.3	If the **cmi.interactions.n.type** is `fill-in,` then the correct response pattern shall be a characterstring value that represents an array of localized_string_types along with properties of the array entries, where each element of the array is separated by a special reserved token [,]. The LMS shall support characterstrings that include at least 10 (the required SPM) localized_string_types. The characterstring shall be represented in the following format: • {case_matters=<boolean>}{order_matters=<boolean>}localized_string_type[,]localized_string_type The following requirements shall be adhered to when building the characterstring:
REQ_91.1.3.1	The correct response pattern array shall consist of 1 or more localized_string_types.
REQ_91.1.3.1.1	The localized_string_type shall have a SPM of 250 characters.
REQ_91.1.3.2	If the correct response pattern has multiple localized_string_types, all of which are required, then each localized_string_type shall be separated by the special reserved token `[,]`.
REQ_91.1.3.3	The case matter's delimiter shall adhere to the following requirements: • the case matters delimiter shall be optionally used in the correct response pattern • the case matters delimiter shall be represented as the reserved token {case_matters=<boolean>}. • the value of <boolean> shall be either `true` (if case matters) or `false` (if case does not matter). • the default value of <boolean> shall be `false` • the case matter delimiter may appear before or after the order matters delimiter
REQ_91.1.3.4	The order matter's delimiter shall adhere to the following requirements: • the order matters delimiter shall be optionally used in the correct response pattern • the order matters delimiter shall be represented as the reserved token {order_matters=<boolean>}. • the value of <boolean> shall be either `true` (if order matters) or `false` (if order does not matter). • the default value of <boolean> shall be `true` • the order matter delimiter may appear before or after the case matters delimiter
REQ_91.1.4	If the **cmi.interactions.n.type** is `long-fill-in,` then the correct response pattern shall be a characterstring value that represents a localized_string_type along with properties of the localized_string_type. The characterstring shall be represented in the following format: • {case_matters=<boolean>}localized_string_type The following requirements shall be adhered to when building the characterstring:
REQ_91.1.4.1	The case matter's delimiter shall adhere to the following requirements: • the case matters delimiter shall be optionally used in the correct response pattern

REQ ID	Requirement
	• the case matters delimiter shall be represented as the reserved token {case_matters=<boolean>}. • the value of <boolean> shall be either `true` (if case matters) or `false` (if case does not matter). • the default value of <boolean> shall be `false`
REQ_91.1.4.2	The localized_string_type shall have a SPM of 4000 characters.
REQ_91.1.5	If the **cmi.interactions.n.type** is `likert`, then the correct response pattern shall be a characterstring representing a valid short_identifier_type.
REQ_91.1.6	If the **cmi.interactions.n.type** is `matching`, then the correct response pattern shall be a characterstring value that represents a bag of <source> / <target> pairs, where each pair is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 36 (the required SPM) pairs. The characterstring shall be represented in the following format: • <source>[.]<target>[,]<source>[.]<target> The following requirements shall be adhered to when building the characterstring:
REQ_91.1.6.1	The <source> and <target> values shall be valid short_identifier_type values.
REQ_91.1.6.2	The special reserved token `[.]` shall be used to separate the <source> from the <target>.
REQ_91.1.6.3	If the correct response pattern has multiple<source> / <target> pairs then each pair shall be separated by the special reserved token `[,]`.
REQ_91.1.7	If the **cmi.interactions.n.type** is `performance`, then the correct response pattern shall be a characterstring value that represents an array of <step_name> and <step_answer> pairs with a property describing the pairs, where each pair is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 125 (the required SPM) pairs. The characterstring shall be represented in the following format: • {order_matters}<step_name>[.]<step_answer>[,]<step_name>[.]<step_answer> The following requirements shall be adhered to when building the characterstring:
REQ_91.1.7.1	The order matter's delimiter shall adhere to the following requirements: • the order matters delimiter shall be optionally used in the correct response pattern • the order matters delimiter shall be represented as the reserved token {order_matters=<boolean>}. • the value of <boolean> shall be either `true` (if order matters) or `false` (if order does not matter). • the default value of <boolean> shall be `true`
REQ_91.1.7.2	The <step_name> shall be represented as a valid short_identifier_type.
REQ_91.1.7.3	The special reserved token `[.]` shall be used to separate the <step_name> from the <step_answer>.
REQ_91.1.7.3.1	The <step_name> may be omitted if there is no step name but only a <step_answer>. In this case, the reserved delimiter `[.]` shall still be present before the <step_answer>.

REQ ID	Requirement
REQ_91.1.7.3.2.	The <step_answer> shall be either a characterstring (SPM of 250 characters) or a numeric.
REQ_91.1.7.3.2.1	The <step_answer> may be omitted if there is no step answer but only a <step_name>. In this case, the reserved delimiter `[.]` shall still be present after the <step_name>.
REQ_91.1.7.3.2.2	If the <step_answer> is a numeric, then the characterstring shall be represented in the following format: • <min>[:]<max> **ADL NOTE:** The <min> represents the lower bound and the <max> represents the upper bound of a range. The <min> and <max> indicate that the correct response is greater than or equal to the <min > and less than or equal to the <max>. If the <min> and <max> are the same, then the correct response is that number.
REQ_91.1.7.3.2.2.1	The <min> and <max> values shall be represented as valid real(10,7)
REQ_91.1.7.3.2.2.2	The <min> may be omitted if there is no lower bound. In this case, the reserved delimiter `[:]` shall still be present before the <max>.
REQ_91.1.7.3.2.2.3	The <max> may be omitted if there is no upper bound. In this case, the reserved delimiter `[:]` shall still be present after the <min>.
REQ_91.1.7.3.2.2.4	Both the <min> and <max> may be omitted if there is no lower or upper bound. In this case, the reserved delimiter `[:]` shall still be present.
REQ_91.1.8	If the **cmi.interactions.n.type** is `sequencing`, then the correct response pattern shall be a characterstring value that represents an array of short_identifier_types, where each element is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 36 (the required SPM) short_identifier_types. The characterstring shall be represented in the following format: • <short_identifier_type>[,]<short_identifier_type> The following requirements shall be adhered to when building the characterstring:
REQ_91.1.8.1	If the correct response pattern is no sequence then the set shall be represented as an empty characterstring (“”). **ADL NOTE:** The empty characterstring implies an empty array; therefore there are no short_identifier_types in the sequence.
REQ_91.1.8.2	If the correct response pattern has multiple short_identifier_types then each short_identifier_type shall be separated by the special reserved token `[,]`.
REQ_91.1.8.3	The order of the short_identifier_types as specified shall be maintained.
REQ_91.1.9	If the **cmi.interactions.n.type** is `numeric`, then the characterstring value shall be represented in the following format: • <min>[:]<max> **ADL NOTE:** The <min> represents the lower bound and the <max> represents the upper bound of a range. The <min> and <max> indicate that the correct response is greater than or equal to the <min > and less than or equal to the <max>. If the <min> and <max> are the same, then the correct response is that number.
REQ_91.1.9.1	The <min> and <max> values shall be represented as valid real(10,7).

REQ ID	Requirement
REQ_91.1.9.2	The <min> may be omitted if there is no lower bound. In this case, the reserved delimiter `[:]` shall still be present before the <max>.
REQ_91.1.9.3	The <max> may be omitted if there is no upper bound. In this case, the reserved delimiter `[:]` shall still be present after the <min>.
REQ_91.1.9.4	Both the <min> and <max> may be omitted if there is no lower or upper bound. In this case, the reserved delimiter `[:]` shall still be present.
REQ_91.1.10	If the **cmi.interactions.n.type** is `other`, then the correct_response shall be a valid characterstring (SPM of 4000 characters).
REQ_91.2	The value of the characterstring for the **cmi.interactions.n.learner_response** shall adhere to the following requirements (dependent on the **cmi.interactions.n.type**).
REQ_91.2.1	If the **cmi.interactions.n.type** is `true-false`, then the characterstring value shall be a characterstring containing one of the following reserved tokens: `true` and `false`.
REQ_91.2.2	If the **cmi.interactions.n.type** is `choice`, then the characterstring value shall be a characterstring value that represents a set of short_identifier_types, where each element of the set is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 36 (the required SPM) short_identifier_types. The characterstring shall be represented in the following format: • <short_identifier_type>[,]<short_identifier_type> The following requirements shall be adhered to when building the characterstring:
REQ_91.2.2.1	If the characterstring is representing a no choice then the set shall be represented as an empty characterstring (“”). **ADL NOTE:** The empty characterstring implies an empty set; therefore there are no short_identifier_types in the set.
REQ_91.2.2.2	If the characterstring has multiple short_identifier_types, then each short_identifier_type shall be separated by the special reserved token `[,]`.
REQ_91.2.2.3	Each short_identifier_type shall be unique within the characterstring.
REQ_91.2.3	If the **cmi.interactions.n.type** is `fill-in`, then the characterstring shall represent an array of localized_string_types, where each element of the array is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 10 (the required SPM) localized_string_types. The characterstring shall be represented in the following format: • localized_string_type[,]localized_string_type The following requirements shall be adhered to when building the characterstring:
REQ_91.2.3.1	The characterstring array shall consist of 1 or more localized_string_types.
REQ_91.2.3.1.1	The localized_string_type shall have a SPM of 250 characters.
REQ_91.2.3.2	If the learner response has multiple localized_string_types, all of which are required, then each localized_string_type shall be separated by the special reserved token `[,]`.
REQ_91.2.4	If the **cmi.interactions.n.type** is `long-fill-in`, then the characterstring shall represent a localized_string_type. The characterstring shall be represented in the following format: • localized_string_type

REQ ID	Requirement
	The following requirements shall be adhered to when building the characterstring:
REQ_91.2.4.1	The characterstring shall consist of 1 and only 1 localized_string_type.
REQ_91.2.4.2	The localized_string_type shall have a SPM of 4000 characters.
REQ_91.2.5	If the **cmi.interactions.n.type** is `likert,` then the characterstring shall represent a valid short_identifier_type.
REQ_91.2.6	If the **cmi.interactions.n.type** is `matching,` then the characterstring shall represent a bag of <source> / <target> pairs, where each pair is separated by a special reserved token `[,].` The LMS shall support characterstrings that include at least 36 (the required SPM) pairs. The characterstring shall be represented in the following format: • <source>[.]<target>[,]<source>[.]<target> The following requirements shall be adhered to when building the characterstring:
REQ_91.2.6.1	The <source> and <target> values shall be valid short_identifier_type values.
REQ_91.2.6.2	The special reserved token `[.]` shall be used to separate the <source> from the <target>.
REQ_91.2.6.3	If the learner response has multiple<source> / <target> pairs then each pair shall be separated by the special reserved token `[,].`
REQ_91.2.7	If the **cmi.interactions.n.type** is `performance,` then the characterstring shall represent an array of <step_name> and <step_answer> pairs, where each pair is separated by a special reserved token `[,].` The LMS shall support characterstrings that include at least 250 (the required SPM) pairs. The characterstring shall be represented in the following format: • <step_name>[.]<step_answer>[,]<step_name>[.]<step_answer> The following requirements shall be adhered to when building the characterstring:
REQ_91.2.7.1	The <step_name> shall be represented as a valid short_identifier_type.
REQ_91.2.7.2	The special reserved token `[.]` shall be used to separate the <step_name> from the <step_answer>.
REQ_91.2.7.2.1	The <step_name> may be omitted if there is no step name but only a <step_answer>. In this case, the reserved delimiter `[.]` shall still be present before the <step_answer>.
REQ_91.2.7.2.2.	The <step_answer> shall be either a characterstring (SPM of 250 characters) or a numeric.
REQ_91.2.7.2.2.1	The <step_answer> may be omitted if there is no step answer but only a <step_name>. In this case, the reserved delimiter `[.]` shall still be present after the <step_name>.
REQ_91.2.7.2.2.2	If the <step_answer> is a numeric, then the characterstring shall be represented in the following format: • <min>[:]<max> **ADL NOTE:** The <min> represents the lower bound and the <max> represents the upper bound of a range. The <min> and <max> indicate that the learner response is greater than or equal to the <min > and less than or equal to the

REQ ID	Requirement
	<max>. If the <min> and <max> are the same, then the learner response is that number.
REQ_91.2.7.2.2.2.1	The <min> and <max> values shall be represented as valid real(10,7)
REQ_91.2.7.2.2.2.2	The <min> may be omitted if there is no lower bound. In this case, the reserved delimiter `[:]` shall still be present before the <max>.
REQ_91.2.7.2.2.2.3	The <max> may be omitted if there is no upper bound. In this case, the reserved delimiter `[:]` shall still be present after the <min>.
REQ_91.2.7.2.2.2.4	Both the <min> and <max> may be omitted if there is no lower or upper bound. In this case, the reserved delimiter `[:]` shall still be present.
REQ_91.2.8	If the **cmi.interactions.n.type** is `sequencing`, then the characterstring shall represent an array of short_identifier_types, where each element is separated by a special reserved token `[,]`. The LMS shall support characterstrings that include at least 36 (the required SPM) short_identifier_types. The characterstring shall be represented in the following format: • <short_identifier_type>[,]<short_identifier_type> The following requirements shall be adhered to when building the characterstring:
REQ_91.2.8.1	If the characterstring needs to represent that there is no sequence then the set shall be represented as an empty characterstring (“”). **ADL NOTE:** The empty characterstring implies an empty array; therefore there are no short_identifier_types in the sequence.
REQ_91.2.8.2	If the learner response has multiple short_identifier_types then each short_identifier_type shall be separated by the special reserved token `[,]`.
REQ_91.2.8.3	The order of the short_identifier_types as specified shall be maintained.
REQ_91.2.9	If the **cmi.interactions.n.type** is `numeric`, then the characterstring shall represent a valid real number, real (10,7).
REQ_91.2.10	If the **cmi.interactions.n.type** is `other`, then the characterstring shall represent a valid characterstring.
REQ_91.2.10.1	The characterstring shall have a SPM of 4000 characters.
REQ_92	All state values shall be represented by reserved characterstring tokens. The actual characterstring tokens are defined by the individual data model elements for which the state is a datatype.

2.1.5. Run-Time Navigation Data Model Conformance Requirements

SCOs can issue certain Navigation Events. These Navigation Events are triggered by a Navigation Request. Navigation Requests are issued by a SCO utilizing the SCORM API and the SCORM Run-Time Navigation Data Model.

LMSs are required to implement all of the SCORM Run-Time Navigation Data Model elements. The following list provides a description of the key terms that are used in the requirements table in this section to describe the LMS Run-Time Navigation Data Model implementation requirements:

- **read-only** – The LMS shall implement this element such that a SCO may only retrieve (read) the value using the GetValue() API method. If the SCO attempts to store (write) a value for this element using the SetValue() API method, the LMS shall behave according to the API Implementation Conformance Requirements (refer to Section 2.1.2).

  For read-only data model elements, SCOs may only invoke the GetValue() API method.

- **read/write** – The LMS shall implement this element such that a SCO may retrieve (read) the value using the GetValue() API method or store (write) the value using the SetValue() API method.

  For read/write data model elements, SCOs may invoke the SetValue() or GetValue() API method.

The LMS shall adhere to the requirements defined in Table 2.1.5a to be considered conformant to the LMS RTE 1.0 conformance category.

Table 2.1.5a: Run-Time Navigation Data Model Conformance Requirements

REQ ID	Requirement
REQ_47	The LMS shall implement the **adl.nav.request** data model element.
REQ_47.1	The **adl.nav.request** data model element shall be implemented as read/write.
REQ_47.2	The **adl.nav.request** data model element shall be implemented as a data model element of type characterstring restricted to the following vocabulary tokens: • `continue` • `previous` • `choice` • `exit` • `exitAll` • `abandon` • `abandonAll` • `_none_`

REQ ID	Requirement
REQ_47.2.1	If the **adl.nav.request** is for a choice navigation request, then the format of the characterstring shall be: `{target=<STRING>}`choice. Where `<STRING>` represents the target of the pending `choice` navigation request.
REQ_47.3	The LMS shall assume the default value of `_none_` for the **adl.nav.request** until the value is set by the SCO.
REQ_47.4	Upon normal termination of a SCO (the SCO set **cmi.exit** to `""` or `normal`) the LMS shall process the navigation request.
REQ_47.5	If the SCO terminates in a suspended state (the SCO set **cmi.exit** to `suspend`), the LMS shall not process the navigation request identified by this element, but should instead process a Suspend or SuspendAll request, as appropriate.
REQ_48	The LMS shall implement the **adl.nav.request_valid.continue** data model element.
REQ_48.1	The **adl.nav.request_valid.continue** data model element shall be implemented as read-only.
REQ_48.2	The **adl.nav.request_valid.continue** data model element shall be implemented to return one of the following restricted vocabulary tokens: • `true` • `false` • `unknown` The LMS shall determine if the request to continue is valid based on the state of the Activity Tree.
REQ_48.3	The default value of the **adl.nav.request_valid.continue** data model element shall be `unknown`.
REQ_49	The LMS shall implement the **adl.nav.request_valid.previous** data model element.
REQ_49.1	The **adl.nav.request_valid.previous** data model element shall be implemented as read-only.
REQ_49.2	The **adl.nav.request_valid.previous** data model element shall be implemented to return one of the following restricted vocabulary tokens: • `true` • `false` • `unknown` The LMS shall determine if the request to go previous is valid based on the state of the Activity Tree.
REQ_49.3	The default value of the **adl.nav.request_valid.previous** data model element shall be `unknown`.
REQ_50	The LMS shall implement the **adl.nav.request_valid.choice.{target=<STRING>}** data model element.
REQ_50.1	The **adl.nav.request_valid.choice.{target=<STRING>}** data model element shall be implemented as read-only.

REQ ID	Requirement
REQ_50.2	The **adl.nav.request_valid.choice.{target=<STRING>}** data model element shall be implemented to return one of the following restricted vocabulary tokens: • `true` • `false` • `unknown` The LMS shall determine if the choice request is valid based on the state of the Activity Tree.
REQ_50.3	The default value of the **adl.nav.request_valid.choice.{target=<STRING>}** data model element shall be `unknown`.

2.1.6. Sequencing Conformance Requirements

LMS conformance to sequencing and navigation is defined by the pseudo code detailed in the SCORM Sequencing and Navigation (SN) [1]. The pseudo code and its behaviors and treatment of data tracking, for all intents and purposes, are the conformance requirements defined for an LMS. How the LMS implements a sequencing implementation is up to an LMS and is treated, from the standpoint of conformance as a "black box" implementation. The SCORM 2004 3rd EditionConformance Test Suite has implemented several test cases for conformance to the sequencing requirements. Each of these test cases has the following information defined:

- An activity tree with associated sequencing rules on different activities in the tree.
- A set of test steps to follow that will produce expected results.

This set of test cases can be found in *Appendix A: Sequencing Conformance Requirements*. SCORM Content Aggregation Packages have been created with the content organization, resources and associated sequencing rules defined.

If an LMS adheres to the test cases and expected behavioral results, then the LMS will be considered conformant to the LMS SN 1.0 conformance category.

2.1.7. User Interface Conformance Requirements

The SCORM 2004 3rd Edition Conformance Test Suite includes User Interface interoperability tests. These tests ensure that LMSs provide learners a minimum set of User Interface devices that would trigger content for delivery, and that LMSs do not provide learners User Interface devices that would trigger pseudo-code exceptions (in most cases). The LMS User Interface tests are implemented throughout the LMS sequencing test cases via inspection. At various times, the tester will be required to inspect the User Interface provided by the LMS and answer several questions. The questions being asked will align with the following definitions and User Interface requirements.

Visible: the existence of a UI device that is either Enabled or Disabled.

- **Enabled**: a UI device that can be triggered by the learner to invoke a given navigation event. In context of the table of contents, "Enabled" is analogous to "Selectable".
- **Disabled**: a UI device that cannot be triggered by the learner to invoke a given navigation event. A non-existent (**Hidden**) UI device is also considered "Disabled".

Hidden: the non-existence of a UI device. The learner shall not be able to invoke the navigation event corresponding to the hidden UI device.

REQ ID	Requirement
REQ_117	An LMS shall provide navigation user interface (UI) devices that support basic (non-exception) navigation through content.
REQ_117.1	An LMS shall provide an enabled navigation UI device corresponding to a Continue Navigation Event when the Current Activity is a child of a (Sequencing Control Mode) Flow equals True cluster. **Note**: This requirement is not currently tested in cases where the "next" activity is either disabled or has a Limit Condition violation, resulting in pseudo-code exception DB.1.1-2.
REQ_117.2	An LMS shall provide a disabled navigation UI device corresponding to a Continue Navigation event when the Current Activity is a child of a (Sequencing Control Mode) Flow equals False cluster.
REQ_117.3	An LMS shall provide an enabled navigation UI device corresponding to a Previous Navigation event when the Current Activity is a child of a (Sequencing Control Mode) Flow equals True and (Sequencing Control Mode) Forward Only equals False cluster; and navigating Previous will not result in walking off the root of the Activity Tree.
REQ_117.4	An LMS shall provide a disabled navigation UI device corresponding to a Previous Navigation event when the Current Activity is a child of a (Sequencing Control Mode) Flow equals False cluster; when the Current Activity is a child of a (Sequencing Control Mode) Forward Only equals True cluster; or if navigating Previous will result in walking off the root of the Activity Tree.
REQ_117.5	An LMS shall provide an enabled "table of contents" UI device (entry in the "table of contents") that provides a means for the learner to trigger a Choice Navigation Event that targets all selectable activities. An activity is selectable in the following circumstances: • It is a leaf activity; its parent has (Sequencing Control Mode) Choice equal to

REQ ID	Requirement
	True; and none of the conditions for being non-selectable (refer to REQ_117.6 and REQ_117.7) apply. • It is a cluster activity; it is either the Root of the Activity Tree or its parent has (Sequencing Control Mode) Choice equal to True; there is a logical "next" (via processing a Continue Navigation Request) activity from it; and none of the conditions for being non-selectable (refer to REQ_117.6 and REQ_117.7) apply.
REQ_117.6	An LMS shall provide a disabled "table of contents" UI device (entry in the "table of contents") that does not provide a means for the learner to trigger a Choice Navigation Event that targets all non-selectable activities. The set of non-selectable activities is defined as follows: • An activity and all of its descendents when the activity has a Disabled Sequencing Rule that evaluates to True. • An activity and all of its descendents when the activity has a Limit Condition violation. • An activity that is a previous sibling of the current activity in a (Sequencing Control Mode) Forward Only equals True cluster. • An activity that is a succeeding sibling of the current activity where that activity or some intervening activity has a Stop Forward Traversal Precondition Rule that evaluates to True. • An activity that is previous in the activity tree relative to the current activity and has a (Sequencing Control Mode) Forward Only equals True cluster preventing the previous traversal. • An activity that is forward in the activity tree relative to the current activity and has a Stop Forward Traversal Precondition Rule that evaluates to True cluster preventing the traversal.
REQ_117.7	An LMS shall not provide TOC UI devices (entries) in the following situations: • Any activity that is "invisible" (described via the isVisible attribute in the CP equal to False). • An activity and all of its descendents when the activity has a Hidden from Choice Sequencing Rule that evaluates to True. • Any activity and all of its descendents when that activity is prevented from being a target of Choice Navigation Event due to: o An (Sequencing Control Mode) Choice Exit equal False ancestor of the Current Activity o Prevent Activation equal to True o Constrained Choice equal to True – relative to the Current Activity.
REQ_118	An LMS shall not provide navigation UI devices corresponding to "hidden" navigation events (described via the <hideLMSUI> elements in the CP).

SECTION 3
Content Package Conformance Requirements

This page intentionally left blank.

3.1. Content Package Conformance Requirements

This section describes the detailed requirements that must be implemented in a Content Package to be SCORM 2004 3rd Editionconformant. A Content Package is required to support the various aspects of the SCORM Run-Time Environment [1] and the SCORM Content Aggregation Model [1].

The conformance requirements for a Content Package are divided into the following sections to address each the different SCORM Content Package Application Profiles:

- Section 3.1.1: Content Package Conformance Requirements
- Section 3.1.2: Content Aggregation Package Manifest Conformance Requirements
- Section 3.1.3: Sequencing Extensions Conformance Requirements
- Section 3.1.4: Navigation and Presentation Conformance Requirements
- Section 3.1.5: Resource Package Manifest Conformance Requirements

The purpose of the Content Package Conformance Test is to verify that a Content Package implements the conformance requirements as outlined in this section. The Content Package Conformance Test is designed to test several conformance categories:

- Conformance Category 1: **CP CAM 1.0** – The Content Package is conformant to the requirements defined in the SCORM Content Aggregation Model [1].

- Conformance Category 2: **CP RTE 1.0** – The Content Package is conformant to the requirements defined in the SCORM Run-Time Environment [1].

The purpose of the Content Package Conformance Test is to verify that a Content Package implements the conformance requirements as outlined in this section. The Content Package Conformance Test verifies both types of SCORM Content Package Application Profiles:

- Resource Packages: A package containing a collection of learning resources with no defined organization (empty organization element <organization/>).
- Content Aggregation Packages: A packaged containing a collection of organized (content organization) learning resources.

3.1.1. Content Package Conformance Requirements

This section defines high-level requirements that both types of SCORM Content Package Application Profiles must meet to be SCORM 2004 3rd Edition conformant. These conformance requirements outline those that are general to content packages as a whole.

The Content Package shall adhere to the requirements defined in Table 3.1.1a to be considered conformant to the CP CAM 1.0 conformance category.

Table 3.1.1a: Content Package Conformance Requirements

REQ ID	Requirement
REQ_28	A SCORM 3004 3rd Edition Content Aggregation Model (CAM) Version 1.0 Conformant (CP CAM 1.0) content package shall adhere to the following requirements:
REQ_28.1	The content package shall contain a manifest named `imsmanifest.xml`
REQ_28.1.1	The `imsmanifest.xml` file shall be placed at the root of the content package.
REQ_28.1.2	The `imsmanifest.xml` instance shall be well-formed.
REQ_28.1.3	The `imsmanifest.xml` instance shall validate against the IMS Content Packaging XML Schema Definition (XSD) - imscp_v1p1.xsd. **ADL Note:** The IMS version for this XSD is IMS CP 1.1.4. This version can be found as the value of the version attribute in the imscp_v1p1.xsd file.
REQ_28.1.4	The `imsmanifest.xml` instance shall validate against the SCORM 2004 3rd Edition Content Packaging Extension XML XSD Version 1.0 – adlcp_v1p3.xsd.
REQ_28.1.5	If the Content Package is a SCORM Content Aggregation Package Application Profile, then the `imsmanifest.xml` shall be conformant to the requirements of a Content Aggregation Package Manifest.
REQ_28.1.6	If the Content Package is a SCORM Resource Application Profile, then the `imsmanifest.xml` shall be conformant to the requirements of a Resource Manifest.
REQ_28.1.7	If the Content Package manifest contains sequencing information as defined by IMS Simple Sequencing Version 1.0, then those extension elements and/or attributes shall validate against the IMS Simple Sequencing XSD Version 1.0 – imsss_v1p0.xsd.
REQ_28.1.8	If the Content Package manifest contains ADL extensions to support sequencing information as defined in the SCORM 2004 3rd Edition CAM, then those extension elements and/or attributes shall validate against the SCORM 2004 3rd Edition Sequencing Extensions XML XSD Version 1.0 – adlseq_v1p3.xsd.
REQ_28.1.9	If the Content Package manifest contains ADL extensions to support navigation information as defined in the SCORM 2004 3rd Edition CAM, then those extension elements and/or attributes shall validate against the SCORM 2004 3rd Edition Navigation Extensions XML XSD Version 1.0 – adlnav_v1p3.xsd.
REQ_28.2	All supporting schemas referenced directly from the `imsmanifest.xml` that are needed to validate the manifest and files referenced by the **<adlcp:location>** element shall be placed at the root of the content package.
REQ_28.3	If the content package is placed into a Package Interchange File (PIF), then the PIF shall be conformant RFC 1951 with archive format PKZIP Version 2.04g (.zip).

REQ ID	Requirement
REQ_28.4	The content package shall contain at least one SCO or Asset (SCORM resources)
REQ_28.5	All learning resources identified as SCOs, shall be SCORM 2004 3rd Edition conformant.
REQ_28.6	All metadata used in the manifest shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).

3.1.2. Content Aggregation Package Manifest Conformance Requirements

The SCORM Content Aggregation Package Application Profile defines a specification for packaging learning resources (for example, Assets and SCOs) with a specific organization, learning context and/or curricular taxonomy. Depending on the curricular taxonomy defined by organizations, the Content Aggregation Package can represent a variety of content aggregations. Content Aggregation Packages can be built for whole courses and for individual pieces of the course (Modules, Chapters, Lesson, etc.).

The Content Aggregation Package's Manifest shall adhere to the requirements defined in Table 3.1.2a to be considered conformant to the CP CAM 1.0 and CP RTE 1.0 conformance categories.

Table 3.1.2a: Content Aggregation Package Manifest Conformance Requirements

REQ ID	Requirement
REQ_30	The Content Aggregation Package's Manifest XML file (`imsmanifest.xml`) shall contain 1 and only 1 root **<manifest>** element.
REQ_30.1	The **<manifest>** element shall contain 1 and only 1 identifier attribute.
REQ_30.1.1	The identifier attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_30.1.2	The **identifier** attribute shall be unique within the Manifest.
REQ_30.2	The **<manifest>** element shall contain 0 or 1 **version** attribute.
REQ_30.2.1	The **version** attribute shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_30.3	The **<manifest>** element shall contain 0 or 1 **xml:base** attribute.
REQ_30.3.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_30.3.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.

REQ ID	Requirement
REQ_30.3.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_30.3.4	The **xml:base** attribute's value shall end with a trailing slash ("/").
REQ_30.4	The child elements of an **<manifest>** element shall exist in the specified order: • **<metadata>** • **<organizations>** • **<resources>** • **<manifest>** • **extension elements** (e.g., **<imsss:sequencingCollection>**) The order is not defined for extension elements, they are only required to be placed after the child **<manifest>** element.
REQ_30.5	The **<manifest>** element shall contain 1 and only 1 **<metadata>** child element.
REQ_30.5.1	The child elements of an **<metadata>** element shall exist in the specified order: • **<schema>** • **<schemaversion>** • **{Metadata}**. The order is not defined for extension elements, they are only required to be placed after the **<schemaversion>** element.
REQ_30.5.2	The **<metadata>** element shall contain 1 and only 1 **<schema>** child element.
REQ_30.5.2.1	The **<schema>** element's value shall be set to the restricted characterstring token: • `ADL SCORM`
REQ_30.5.3	The **<metadata>** element shall contain 1 and only 1 **<schemaversion>** child element.
REQ_30.5.3.1	The **<schemaversion>** element's value shall be set to the restricted characterstring token: • `2004 3rd Edition`
REQ_30.5.4	The **<metadata>** element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the **<adlcp:location>** element to reference a stand-alone XML instance document.
REQ_30.5.4.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.5.4.1.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the content aggregation.
REQ_30.5.4.1.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the content aggregation. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_30.5.4.1.2.1	The value of this element shall be a characterstring that represents the URL (as

REQ ID	Requirement
	defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_30.6	The <**manifest**> element shall contain 1 and only 1 <**organizations**> child element.
REQ_30.6.1	The <**organizations**> element shall have 1 and only 1 **default** attribute.
REQ_30.6.1.1	The **default** attribute's value shall be a valid identifier of the default <**organization**> in the manifest. The value shall reference an <**organization**> element that is a direct descendant of the <**organizations**> element.
REQ_30.6.1.2	The **default** attribute's value shall be represented as an xs:IDREF type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:IDREF.
REQ_30.6.2	The child elements of an <**organizations**> element shall exist in the specified order: • <**organization**> • **extension elements** The order is not defined for extension elements, they are only required to be placed after the <**organization**> element.
REQ_30.6.3	The <**organizations**> element shall contain 1 or More <**organization**> child elements.
REQ_30.6.3.1	The <**organization**> element shall contain 1 and only 1 **identifier** attribute.
REQ_30.6.3.1.1	The **identifier** attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_30.6.3.1.2	The **identifier** attribute shall be unique within the Manifest.
REQ_30.6.3.2	The <**organization**> element shall contain 0 or 1 **structure** attribute.
REQ_30.6.3.2.1	The **structure** attribute's value shall be a characterstring representing the structure of the organization. The default value, if the attribute is not provided, is `hierarchical`.
REQ_30.6.3.3	The <**organization**> element shall contain 0 or 1 **adlseq:objectivesGlobalToSystem** attribute.
REQ_30.6.3.3.1	The **adlseq:objectivesGlobalToSystem** attribute shall be a boolean value of either `true` or `false`.
REQ_30.6.3.4	The child elements of an <**organization**> element, if used, shall exist in the specified order: • <**title**> • <**item**> • <**metadata**> • **extension elements** (e.g., <**imsss:sequencing**>) The order is not defined for extension elements, they are only required to be placed after the <**metadata**> element.
REQ_30.6.3.5	The <**organization**> element shall contain 1 and only 1 <**title**> child element.
REQ_30.6.3.5.1	The <**title**> element's value shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of

REQ ID	Requirement
	values of an xs:string.
REQ_30.6.3.6	The **<organization>** element shall contain 1 or More **<item>** child elements.
REQ_30.6.3.6.1	The **<item>** element shall contain 1 and only 1 **identifier** attribute.
REQ_30.6.3.6.1.1	The **identifier** attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_30.6.3.6.1.2	The **identifier** attribute shall be unique within the Manifest.
REQ_30.6.3.6.2	The **<item>** element shall contain 0 or 1 **identifierref** attribute.
REQ_30.6.3.6.2.1	The **identifierref** attribute's value shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_30.6.3.6.2.2	If used, the **identifierref** attribute shall match either an **identifier** attribute of a **<resource>** in the current Manifest, the **identifier** attribute of a **<resource>** in a (sub)manifest, an **identifier** attribute of **<manifest>** ((sub)manifest) in the scope of the current Manifest. **ADL Note:** The IMS Global Consortium, Inc., is working on a new version of the IMS Content Packaging Specification. One of the major issues that IMS is resolving deals with (sub)manifests, their use, requirements of use and XML syntax requirements. At this time, ADL recommends not to use (sub)manifests until completion of the IMS work. Any questions, concerns or further recommendations on (sub)manifests should be sent to ADL
REQ_30.6.3.6.2.3	The **identifierref** shall not be used on **<item>** elements that contain other **<item>** elements.
REQ_30.6.3.6.2.4	A leaf **<item>** shall contain an **identifierref** attribute. **ADL Note:** The identifierrf attribute shall adhere to the requirements defined in REQ_30.6.3.6.2.2.
REQ_30.6.3.6.3	The **<item>** element shall contain 0 or 1 **isvisible** attribute.
REQ_30.6.3.6.3.1	The **isvisible** attribute shall be a characterstring representation of the boolean values: true false
REQ_30.6.3.6.4	The **<item>** element shall contain 0 or 1 **parameters** attribute.
REQ_30.6.3.6.4.1	The **parameters** attribute's value shall be a valid characterstring.
REQ_30.6.3.6.4.2	The **parameters** attribute's value syntax shall be limited to: • #<parameter> • <name>=<value>(&<name>=<value>)*(#<parameter>) • ?<name>=<value>(&<name>=<value>)*(#<parameter>) where: • <parameter>, <name> and <value> is some implementation defined characterstring value • = is required to separate the <name> and <value> pair

REQ ID	Requirement
	• & is required to separate multiple sets of <name> and <value> pairs • (&<name>=<value>)* indicates that 0 or more <name> and <value> pairs can be concatenated together ADL Note: NOTE: The characters used in the parameters value may need to be URL encoded. RFC 3986 defines the requirements for encoding URLs.
REQ_30.6.3.6.5	The child elements of an **<item>** element, if used, shall exist in the specified order: • **<title>** • **<item>** • **<metadata>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<metadata>** element).
REQ_30.6.3.6.6	The **<item>** element shall contain 1 and only 1 **<title>** child element.
REQ_30.6.3.6.6.1	The **<title>** element's value shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_30.6.3.6.7	The **<item>** element shall contain 0 or More **<item>** child elements.
REQ_30.6.3.6.7.1	The **<item>** element can be nested an arbitrary number of depths, each nested **<item>** element shall adhere to those **<item>** element requirements defined above.
REQ_30.6.3.6.8	The **<item>** element shall contain 0 or 1 **<metadata>** child element.
REQ_30.6.3.6.8.1	The **<metadata>** element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the **<adlcp:location>** element to reference a stand-alone XML instance document.
REQ_30.6.3.6.8.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.6.3.6.8.1.1.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the Activity.
REQ_30.6.3.6.8.1.1.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the Activity. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_30.6.3.6.8.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_30.6.3.6.9	The **<item>** element shall contain 0 or 1 **<adlcp:timeLimitAction>** child element.

REQ ID	Requirement
REQ_30.6.3.6.9.1	The <**adlcp:timeLimitAction**> element shall only be used with <**item**> elements that references SCO resources.
REQ_30.6.3.6.9.2	The <**adlcp:timeLimitAction**> element's value shall be one of the following restricted characterstrings: • `exit,message` • `exit,no message` • `continue,message` • `continue,no message`
REQ_30.6.3.6.10	The <**item**> element shall contain 0 or 1 <**adlcp:dataFromLMS**> child element.
REQ_30.6.3.6.10.1	The <**adlcp:dataFromLMS**> element shall only be used with <**item**> elements that references SCO resources.
REQ_30.6.3.6.10.2	The <**adlcp:dataFromLMS**> element's value shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string
REQ_30.6.3.6.11	The <**item**> element shall contain 0 or 1 <**imsss:sequencing**> child elements. Refer to *Section 3.1.3 Sequencing Extension Conformance Requirements* for more information.
REQ_30.6.3.6.12	The <**item**> element shall contain 0 or 1 <**adlnav:presentation**> child element. Refer to *Section 3.1.4 Navigation and Presentation Extensions Conformance Requirements* for more information.
REQ_30.6.3.6.13	The <**item**> element shall contain 0 or 1 <**adlcp:completionThreshold**> child element.
REQ_30.6.3.6.13.1	The <**adlcp:completionThreshold**> element shall only be used with <**item**> elements that reference SCO resources.
REQ_30.6.3.6.13.2	The <**adlcp:completionThreshold**> element's value shall be represented as an xs:decimal value that falls in the range of (0.0..1.0), inclusive. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string
REQ_30.6.3.7	The <**organization**> element shall contain 0 or 1 <**metadata**> child element.
REQ_30.6.3.7.1	The <**metadata**> element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the <**adlcp:location**> element to reference a stand-alone XML instance document.
REQ_30.6.3.7.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.6.3.7.1.1.1	The <**metadata**> element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the content organization.
REQ_30.6.3.7.1.1.2	The <**metadata**> element shall contain 0 or More <**adlcp:location**> child elements to define the location (URL) of the **{Metadata}** describing the content organization. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if

REQ ID	Requirement
	provided.
REQ_30.6.3.7.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_30.6.3.8	The **<organization>** element shall contain 0 or 1 **<imsss:sequencing>** child elements. Refer to the *Section 3.1.3 Sequencing Extensions Conformance Requirements.*
REQ_30.7	The **<manifest>** element shall contain 1 and only 1 **<resources>** child element.
REQ_30.7.1	The **<resources>** element shall contain 0 or 1 **xml:base** attribute.
REQ_30.7.1.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_30.7.1.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_30.7.1.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_30.7.1.4	The **xml:base** attribute's value shall end with a trailing slash ("/").
REQ_30.7.2	The child elements of an **<resources>** element shall exist in the specified order: • **<resource>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<resource>** element.
REQ_30.7.3	The **<resources>** element shall contain 0 or More **<resource>** elements.
REQ_30.7.3.1	The **<resource>** element shall contain 1 and only 1 **identifier** attribute.
REQ_30.7.3.1.1	The **identifier** attribute's value shall be a valid xs:ID. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_30.7.3.1.2	The **identifier** attribute's value shall be unique within the Manifest.
REQ_30.7.3.2	The **<resource>** element shall contain 1 and only 1 **type** attribute.
REQ_30.7.3.2.1	The **type** attribute's value shall be represented as an xs:string type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_30.7.3.3	The **<resource>** element shall contain 0 or 1 **href** attribute.
REQ_30.7.3.3.1	The **href** attribute shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax) for the resource. **ADL NOTE:** The **href** attribute is affected by the **xml:base** attribute if provided.
REQ_30.7.3.3.2	The **href** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.

REQ ID	Requirement
REQ_30.7.3.3.3	The **href** attribute's value shall not begin with a leading slash ("/").
REQ_30.7.3.4	The <**resource**> element shall contain 1 and only 1 **adlcp:scormType** attribute.
REQ_30.7.3.4.1	The **adlcp:scormType** value shall be a characterstring where the characterstring is either: • `sco` if the resource is a SCO, or • `asset` if the resource is an Asset.
REQ_30.7.3.5	The <**resource**> element shall contain 0 or 1 **xml:base** attribute.
REQ_30.7.3.5.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_30.7.3.5.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_30.7.3.5.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_30.7.3.5.4	The **xml:base** attribute's value shall end with a trailing slash ("/").
REQ_30.7.3.7	The child elements of an <**resource**> element shall exist in the specified order: • <**metadata**> • <**file**> • <**dependency**> • **extension elements** The order is not defined for extension elements, they are only required to be placed after the <**dependency**> element.
REQ_30.7.3.8	The <**resource**> element shall contain 0 or 1 <**metadata**> child element.
REQ_30.7.3.8.1	The <**metadata**> element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the <**adlcp:location**> element to reference a stand-alone XML instance document.
REQ_30.7.3.8.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.7.3.8.1.1.1	The <**metadata**> element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the Asset resource.
REQ_30.7.3.8.1.1.2	The <**metadata**> element shall contain 0 or More <**adlcp:location**> child elements to define the location (URL) of the **{Metadata}** describing the Asset resource. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_30.7.3.8.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).

REQ ID	Requirement
REQ_30.7.3.8.1.2	If providing Metadata, the **{Metadata}** be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.7.3.8.1.2.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the SCO resource.
REQ_30.7.3.8.1.2.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the SCO resource. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_30.7.3.8.1.2.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_30.7.3.9	The **<resource>** element shall contain 0 or More **<file>** child elements.
REQ_30.7.3.9.1	The **<file>** element shall identify 0 or More local files that this resource is dependent on. **ADL NOTE:** For all files that are required for delivery and are local to the content package (physically located in a content package), a **<file>** element shall be used to represent the file relative to the resource in which it is used.
REQ_30.7.3.9.1.1	If the **<resource>** is local to the content package, then a **<file>** element must exist as a child of the defined **<resource>** element and the **<file>** element's **href** attribute shall be identical to the **<resource>** element's **href** attribute, exclusive of any URL parameters that may be specified in the **href** attribute of the **<resource>** element.
REQ_30.7.3.9.2	The **<file>** element shall contain 1 and only 1 **href** attribute.
REQ_30.7.3.9.2.1	The **href** attribute shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax) for the resource. **ADL NOTE:** The **href** attribute is affected by the **xml:base** attribute if provided.
REQ_30.7.3.9.2.2	The **href** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_30.7.3.9.2.3	The **href** attribute's value shall not begin with a leading slash ("/").
REQ_30.7.3.9.3	The child elements of an **<file>** element shall exist in the specified order: • **<metadata>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<metadata>** element.
REQ_30.7.3.9.4	The **<file>** element shall contain 0 or 1 **<metadata>** child element.

REQ ID	Requirement
REQ_30.7.3.9.4.1	The **<metadata>** element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the **<adlcp:location>** element to reference a stand-alone XML instance document.
REQ_30.7.3.9.4.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_30.7.3.9.4.1.1.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the Asset.
REQ_30.7.3.9.4.1.1.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the Asset. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_30.7.3.9.4.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_30.7.3.10	The **<resource>** element shall contain 0 or More **<dependency>** child elements.
REQ_30.7.3.10.1	The **<dependency>** element shall contain 1 and only 1 **identifierref** attribute.
REQ_30.7.3.10.1.1	The **identifierref** attribute's value shall be an xs:IDRef. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:IDRef.
REQ_30.7.3.10.1.2	The **identifierref** attribute's value shall reference an **identifier** attribute of a **<resource>** (within the scope of the **<manifest>** element for which it is defined).
REQ_30.7.3.10.2	The **<dependency>** element shall be an empty XML element (i.e., **<dependency/>**. **ADL NOTE:** An empty element, by definition, is permitted to contain attributes.
REQ_30.8	The **<manifest>** element (root) shall contain 0 or More **<manifest>** child elements, i.e., (sub)manifests. **ADL Note:** The IMS Global Consortium Inc., is working on a new version of the IMS Content Packaging Specification. One of the major revisions that IMS is resolving deals with (sub)manifests, their use, requirements of use and XML syntax requirements. At this time, ADL recommends not to use (sub)manifests until completion of the IMS work.
REQ_30.9	The **<manifest>** element shall contain 0 or 1 **<imsss:sequencingCollection>** child element. Refer to *Section 3.1.3 Sequencing Extension Conformance Requirements*.

3.1.3. Sequencing Extensions Conformance Requirements

The SCORM SN defines the requirements that must be met when encoding specific sequencing and navigation strategies in a Content Aggregation Package Application Profile Manifest. The elements listed in Table 3.1.3a are optional. If a content package manifest uses the sequencing elements defined in Table 3.1.3a, the manifest shall conform to the requirements defined below.

Table 3.1.3a: Sequencing Extensions Conformance Requirements

REQ ID	Requirement
REQ_31	The <**sequencing**> element shall adhere to the following requirements.
REQ_31.1	The <**sequencingCollection**> element shall contain 1 or More <**sequencing**> elements.
REQ_31.2	The <**item**> and <**organization**> elements shall contain 0 or 1 <**sequencing**> element.
REQ_31.2.1	If a leaf <**item**> element references a (sub)manifest, that leaf <**item**> element shall not have a <**sequencing**> element as a child. **ADL Note:** The IMS Global Consortium, Inc., is working on a new version of the IMS Content Packaging Specification. One of the major issues that IMS is resolving deals with (sub)manifests, their use, requirements of use and XML syntax requirements. At this time, ADL recommends not to use (sub)manifests until completion of the IMS work. Any questions, concerns or further recommendations on (sub)manifests should be sent to ADL
REQ_31.3	The <**sequencing**> element shall contain 0 or 1 **ID** attribute.
REQ_31.3.1	The **ID** attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_31.3.2	The **ID** attribute shall be unique within the Manifest.
REQ_31.3.3	If the <**sequencing**> element is a child of a <**sequencingCollection**> element, the **ID** attribute is mandatory (1 and only 1 time).
REQ_31.3.4	If the <**sequencing**> element is a child of an <**item**> or <**organization**> element, then the **ID** attribute is not permitted.
REQ_31.4	The <**sequencing**> element shall contain 0 or 1 **IDRef** attribute.
REQ_31.4.1	The **IDRef** attribute's value shall be represented as an xs:IDRef type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:IDREF.
REQ_31.4.2	The **IDRef** attribute shall be unique within the Manifest.
REQ_31.4.3	If the <**sequencing**> element is defined to reference a <**sequencing**> element defined in the <**sequencingCollection**> then the <**sequencing**> element shall have an **IDRef** attribute value that matches a <**sequencing**> element's **ID** attribute found in the <**sequencingCollection**>.
REQ_31.5	The <**sequencing**> element shall contain 0 or 1 <**controlMode**> child elements.
REQ_31.5.1	The <**controlMode**> element shall contain 0 or 1 **choice** attribute.

REQ ID	Requirement
REQ_31.5.1.1	The **choice** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.1.2	The **choice** attribute's value shall be either: • `true` (default if no choice attribute is present), or • `false`
REQ_31.5.2	The <**controlMode**> element shall contain 0 or 1 **choiceExit** attribute.
REQ_31.5.2.1	The **choiceExit** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.2.2	The **choiceExit** attribute's value shall be either: • `true` (default if no choiceExit attribute is present), or • `false`
REQ_31.5.3	The <**controlMode**> element shall contain 0 or 1 **flow** attribute.
REQ_31.5.3.1	The **flow** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.3.2	The **flow** attribute's value shall be either: • `true`, or • `false` (default if no flow attribute is present)
REQ_31.5.4	The <**controlMode**> element shall contain 0 or 1 **forwardOnly** attribute.
REQ_31.5.4.1	The **forwardOnly** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.4.2	The **forwardOnly** attribute's value shall be either : • `true`, or • `false` (default if no flow attribute is present)
REQ_31.5.5	The <**controlMode**> element shall contain 0 or 1 **useCurrentAttemptObjectiveInfo** attribute.
REQ_31.5.5.1	The **useCurrentAttemptObjectiveInfo** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.5.2	The **useCurrentAttemptObjectiveInfo** attribute's value shall be either: • `true` (default if no choice attribute is present), or • `false`
REQ_31.5.6	The <**controlMode**> element shall contain 0 or 1 **useCurrentAttemptProgressInfo** attribute.

REQ ID	Requirement
REQ_31.5.6.1	The **useCurrentAttemptProgressInfo** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.5.6.2	The **useCurrentAttemptProgressInfo** attribute's value shall be either: • `true` (default if no choice attribute is present), or • `false`
REQ_31.6	The <**sequencing**> element shall contain 0 or 1 <**sequencingRules**> child element.
REQ_31.6.1	The <**sequencingRules**> element shall contain 0 or More <**preConditionRule**> child elements.
REQ_31.6.1.1	The <**preConditionRule**> element shall contain 1 and only 1 <**ruleConditions**> child elements.
REQ_31.6.1.1.1	The <**ruleConditions**> element shall contain 0 or 1 **conditionCombination** attribute.
REQ_31.6.1.1.1.1	The **conditionCombination** attribute's value shall be one of the following restricted tokens: • `all` (default if no **conditionCombination** is provided) • `any`
REQ_31.6.1.1.2	The <**ruleConditions**> element shall contain 1 or More <**ruleCondition**> child elements. **ADL NOTE:** If the <**ruleConditions**> element is empty or not declared in a sequencing rule, the sequencing rule shall be ignored by a conforming implementation.
REQ_31.6.1.1.2.1	The <**ruleCondition**> element shall contain 0 or 1 **referencedObjective** attribute.
REQ_31.6.1.1.2.1.1	If the **referencedObjective** attribute is used, the value shall be an xs:string and the value shall represent an identifier of an objective found within the Manifest. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_31.6.1.1.2.1.2	Since the underlying data type of a **referencedObjective** is a unique identifier, then the value held by the **referencedObjective** shall not be an empty characterstring nor contain all white space characters.
REQ_31.6.1.1.2.1.3	If provided, the value of the **referencedObjective** shall contain an **objectiveID** of either the <**primaryObjective**> or an <**objective**> elemented defined for the activity.
REQ_31.6.1.1.2.2	The <**ruleCondition**> element shall contain 0 or 1 **measureThreshold** attribute.
REQ_31.6.1.1.2.2.1	The **measureThreshold** attribute's value shall be a valid xs:decimal in the range of -1.0000 and 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal
REQ_31.6.1.1.2.2.2	If no **measureThreshold** is provided then the value shall be 0.0 (default).

REQ ID	Requirement
REQ_31.6.1.1.2.3	The <**ruleCondition**> element shall contain 0 or 1 **operator** attribute.
REQ_31.6.1.1.2.3.1	The **operator** attribute's value shall be one of the following restricted tokens: • `not` • `noOp` (default if the attribute is not provided)
REQ_31.6.1.1.2.4	The <**ruleCondition**> element shall contain 1 and only 1 **condition** attribute.
REQ_31.6.1.1.2.4.1	The **condition** attribute's value shall be one of the following restricted tokens: • `satisfied` • `objectiveStatusKnown` • `objectiveMeasureKnown` • `objectiveMeasureGreaterThan` • `objectiveMeasureLessThan` • `completed` • `activityProgressKnown` • `attempted` • `attemptLimitExceeded` • `timeLimitExceeded` • `outsideAvailableTimeRange` • `always` **ADL NOTE:** If no **condition** attribute is provided the action defined for the condition is always taken.
REQ_31.6.1.2	The <**preConditionRule**> shall contain 1 and only 1 <**ruleAction**> child element.
REQ_31.6.1.2.1	The <**ruleAction**> element shall contain 1 and only 1 **action** attribute.
REQ_31.6.1.2.1.1	The **action** attribute's value shall be one of the following restricted tokens: • `skip` • `disabled` • `hiddenFromChoice` • `stopForwardTraversal`
REQ_31.6.2	The <**sequencingRules**> element shall contain 0 or More <**exitConditionRule**> child elements.
REQ_31.6.2.1	The <**exitConditionRule**> element shall contain 1 and only 1 <**ruleConditions**> child elements.
REQ_31.6.2.1.1	The <**ruleConditions**> element shall contain 0 or 1 **conditionCombination** attribute.
REQ_31.6.2.1.1.1	The **conditionCombination** attribute's value shall be one of the following restricted tokens: • `all` (value if no conditionCombination is provide) • `any`

REQ ID	Requirement
REQ_31.6.2.1.2	The **<ruleConditions>** element shall contain 1 or More **<ruleCondition>** child elements. **ADL NOTE:** If the **<ruleConditions>** element is empty or not declared in a sequencing rule, the sequencing rule shall be ignored by a conforming implementation
REQ_31.6.2.1.2.1	The **<ruleCondition>** element shall contain 0 or 1 **referencedObjective** attribute.
REQ_31.6.2.1.2.1.1	If the **referencedObjective** attribute is used, the value shall be an xs:string and the value shall represent an identifier of an objective found within the Manifest. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_31.6.2.1.2.1.2	Since the underlying data type of a **referencedObjective** is a unique identifier, then the value held by the **referencedObjective** shall not be an empty characterstring nor contain all white space characters.
REQ_31.6.2.1.2.1.3	The value of the **referencedObjective** shall contain an **objectiveID** of either the **<primaryObjective>** or an **<objective>** element defined for the activity.
REQ_31.6.2.1.2.2	The **<ruleCondition>** element shall contain 0 or 1 **measureThreshold** attribute.
REQ_31.6.2.1.2.2.1	The **measureThreshold** attribute's value shall be a valid xs:decimal in the range of -1.0000 and 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal
REQ_31.6.2.1.2.2.2	If no **measureThreshold** is provided then the value shall be 0.0 (default).
REQ_31.6.2.1.2.3	The **<ruleCondition>** element shall contain 0 or 1 **operator** attribute.
REQ_31.6.2.1.2.3.1	The **operator** attribute's value shall be one of the following restricted tokens: • `not` • `noOp` (default if the attribute is not provided)
REQ_31.6.2.1.2.4	The **<ruleCondition>** element shall contain 1 and only 1 **condition** attribute.
REQ_31.6.2.1.2.4.1	The **condition** attribute's value shall be one of the following restricted tokens: • `satisfied` • `objectiveStatusKnown` • `objectiveMeasureKnown` • `objectiveMeasureGreaterThan` • `objectiveMeasureLessThan` • `completed` • `activityProgressKnown` • `attempted` • `attemptLimitExceeded` • `timeLimitExceeded` • `outsideAvailableTimeRange` • `always` **ADL NOTE:** If no **condition** attribute is provided the **action** defined for the condition is always taken.

REQ ID	Requirement
REQ_31.6.2.2	The <**exitConditionRule**> shall contain 1 and only 1 <**ruleAction**> child elements.
REQ_31.6.2.2.1	The <**ruleAction**> element shall contain 1 and only 1 **action** attribute.
REQ_31.6.2.2.1.1	The **action** attribute's value shall be the following restricted token: • `exit`
REQ_31.6.3	The <**sequencingRules**> element shall contain 0 or More <**postConditionRule**> child elements.
REQ_31.6.3.1	The <**postConditionRule**> element shall contain 1 and only 1 <**ruleConditions**> child elements.
REQ_31.6.3.1.1	The <**ruleConditions**> element shall contain 0 or 1 **conditionCombination** attribute.
REQ_31.6.3.1.1.1	The **conditionCombination** attribute's value shall be one of the following restricted tokens: • `all` (default if no **conditionCombination** is provide) • `any`
REQ_31.6.3.1.2	The <**ruleConditions**> element shall contain 1 or More <**ruleCondition**> child elements. **ADL NOTE:** If the <**ruleConditions**> element is empty or not declared in a sequencing rule, the sequencing rule shall be ignored by a conforming implementation
REQ_31.6.3.1.2.1	The <**ruleCondition**> element shall contain 0 or 1 **referencedObjective** attribute.
REQ_31.6.3.1.2.1.1	If the **referencedObjective** attribute is used, the value shall be an xs:string and the value shall represent an identifier of an objective found within the Manifest. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_31.6.3.1.2.1.2	Since the underlying data type of a **referencedObjective** is a unique identifier, then the value held by the **referencedObjective** shall not be an empty characterstring nor contain all white space characters.
REQ_31.6.3.1.2.1.3	The value of the **referencedObjective** shall contain an **objectiveID** of either the <**primaryObjective**> or an <**objective**> element defined for the activity.
REQ_31.6.3.1.2.2	The <**ruleCondition**> element shall contain 0 or 1 **measureThreshold** attribute.
REQ_31.6.3.1.2.2.1	The **measureThreshold** attribute's value shall be a valid xs:decimal in the range of -1.0000 and 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal.
REQ_31.6.3.1.2.2.2	If no **measureThreshold** is provided then the value shall be 0.0 (default).
REQ_31.6.3.1.2.3	The <**ruleCondition**> element shall contain 0 or 1 **operator** attribute.
REQ_31.6.3.1.2.3.1	The **operator** attribute's value shall be one of the following restricted tokens: • `not` • `noOp` (default if the attribute is not provided).

REQ ID	Requirement
REQ_31.6.3.1.2.4	The <**ruleCondition**> element shall contain 1 and only 1 **condition** attribute.
REQ_31.6.3.1.2.4.1	The **condition** attribute's value shall be one of the following restricted tokens: • `satisfied` • `objectiveStatusKnown` • `objectiveMeasureKnown` • `objectiveMeasureGreaterThan` • `objectiveMeasureLessThan` • `completed` • `activityProgressKnown` • `attempted` • `attemptLimitExceeded` • `timeLimitExceeded` • `outsideAvailableTimeRange` • `always` **ADL NOTE**: If no **condition** attribute is provided the **action** defined for the condition is always taken.
REQ_31.6.3.2	The <**postConditionRule**> shall contain 1 and only 1 <**ruleAction**> child elements.
REQ_31.6.3.2.1	The <**ruleAction**> element shall contain 1 and only 1 **action** attribute.
REQ_31.6.3.2.1.1	The **action** attribute's value shall be one of the following restricted tokens: • `exitParent` • `exitAll` • `retry` • `retryAll` • `continue` • `previous`
REQ_31.7	The <**sequencing**> element shall contain 0 or 1 <**limitConditions**> child element.
REQ_31.7.1	The <**limitConditions**> element shall contain 0 or 1 **attemptLimit** attribute.
REQ_31.7.1.1	The **attemptLimit** attribute's value shall be a non-negative integer. **ADL NOTE:** The default value is 0 if the **attemptLimit** attribute is not present.
REQ_31.7.2	The <**limitConditions**> element shall contain 0 or 1 **attemptAbsoluteDurationLimit** attribute.
REQ_31.7.2.1	The **attemptAbsoluteDurationLimit** attribute's value shall be a valid xs:duration datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:duration.
REQ_31.7.3	The <**limitConditions**> element shall contain 0 or 1 **attemptExperiencedDurationLimit** attribute.

REQ ID	Requirement
REQ_31.7.3.1	The **attemptExperiencedDurationLimit** attribute's value shall be a valid xs:duration datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:duration. **ADL NOTE:** At this time, the SCORM does not require an LMS to implement time related sequencing decisions based on this value.
REQ_31.7.4	The **<limitConditions>** element shall contain 0 or 1 **activityAbsoluteDurationLimit** attribute.
REQ_31.7.4.1	The **activityAbsoluteDurationLimit** attribute's value shall be a valid xs:duration datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:duration. **ADL NOTE:** At this time, the SCORM does not require an LMS to implement time related sequencing decisions based on this value.
REQ_31.7.5	The **<limitConditions>** element shall contain 0 or 1 **activityExperiencedDurationLimit** attribute.
REQ_31.7.5.1	The **activityExperiencedDurationLimit** attribute's value shall be a valid xs:duration datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:duration. **ADL NOTE**: At this time, the SCORM does not require an LMS to implement time related sequencing decisions based on this value.
REQ_31.7.6	The **<limitConditions>** element shall contain 0 or 1 **beginTimeLimit** attribute.
REQ_31.7.6.1	The **beginTimeLimit** attribute's value shall be a valid xs:dateTime datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:dateTime. **ADL NOTE**: At this time, the SCORM does not require an LMS to implement time related sequencing decisions based on this value.
REQ_31.7.7	The **<limitConditions>** element shall contain 0 or 1 **endTimeLimit** attribute.
REQ_31.7.7.1	The **endTimeLimit** attribute's value shall be a valid xs:dateTime datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:dateTime. **ADL NOTE:** At this time, the SCORM does not require an LMS to implement time related sequencing decisions based on this value.
REQ_31.8	The **<sequencing>** element shall contain 0 or 1 **<auxiliaryResources>** child element. **ADL NOTE:** At this time, the SCORM does not require an LMS to support auxiliary resources.
REQ_31.8.1	The **<auxiliaryResources>** element shall contain 0 or 1 **auxiliaryResourceID** attribute. **ADL NOTE:** At this time, the SCORM does not require an LMS to support auxiliary resources.

REQ ID	Requirement
REQ_31.8.1.1	The **auxiliaryResourceID** attribute's value shall be represented as an xs:anyURI. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_31.8.2	The <**auxiliaryResources**> element shall contain 0 or 1 **purpose** attribute. **ADL NOTE:** At this time, the SCORM does not require an LMS to support auxiliary resources.
REQ_31.8.2.1	The **purpose** attribute's value shall be represented as an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_31.9	The <**sequencing**> element shall contain 0 or 1 <**rollupRules**> child element.
REQ_31.9.1	The <**rollupRules**> element shall contain 0 or 1 **rollupObjectiveSatisfied** attribute.
REQ_31.9.1.1	The **rollupObjectiveSatisfied** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.9.1.2	The **rollupObjectiveSatisfied** attribute's value shall be one of the following restricted tokens: • `true` (default if no **rollupObjectiveSatisfied** attribute is present) • `false`
REQ_31.9.2	The <**rollupRules**> element shall contain 0 or 1 **rollupProgressCompletion** attribute.
REQ_31.9.2.1	The **rollupProgressCompletion** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean
REQ_31.9.2.2	The **rollupProgressCompletion** attribute's value shall be one of the following restricted tokens: • `true` (default if no **rollupProgressCompletion** attribute is present) • `false`
REQ_31.9.3	The <**rollupRules**> element shall contain 0 or 1 **objectiveMeasureWeight** attribute.
REQ_31.9.3.1	The **objectiveMeasureWeight** attribute's value shall be represented as an xs:decimal in the range of 0.0000 to 1.0000 (precision to at least 4 significant decimal positions). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal.
REQ_31.9.4	The <**rollupRules**> element shall contain 0 or More <**rollupRule**> child elements.
REQ_31.9.4.1	The <**rollupRule**> element shall contain 0 or 1 **childActivitySet** attribute.
REQ_31.9.4.1.1	The **childActivitySet** attribute's value shall be one of the following restricted tokens: • `all` (default if no **childActivitySet** attribute is defined)

REQ ID	Requirement
	• any • none • atLeastCount • atLeastPercent
REQ_31.9.4.2	The <**rollupRule**> element shall contain 0 or 1 **minimumCount** attribute.
REQ_31.9.4.2.1	The **minimumCount** attribute's value shall be a non-negative integer (default is 0 if no **minimumCount** attribute is defined).
REQ_31.9.4.2.2	The **minimumCount** attribute shall be used if the **childActivitySet** attribute is set to a value of `atLeastCount`. If the **minimumCount** attribute is not used, the default value of 0 shall be used.
REQ_31.9.4.3	The <**rollupRule**> element shall contain 0 or 1 **minimumPercent** attribute.
REQ_31.9.4.3.1	The **minimumPercent** attribute's value shall be an xs:decimal in the range of 0.0000 to 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal.
REQ_31.9.4.3.2	The **minimumPercent** attribute shall be used if the **childActivitySet** attribute is set to a value of `atLeastPercent`. If the **minimumPercent** attribute is not used, the default value of 0.0000 shall be used.
REQ_31.9.4.4	The <**rollupRule**> element shall contain 1 and only 1 <**rollupConditions**> child elements.
REQ_31.9.4.4.1	The <**rollupConditions**> element shall contain 0 or 1 **conditionCombination** attribute.
REQ_31.9.4.4.1.1	The **conditionCombination** attribute's value shall be one of the following restricted tokens: • all • any (default if no **conditionCombination** is provided)
REQ_31.9.4.4.2	The <**rollupConditions**> element shall contain 1 or More <**rollupCondition**> child elements.
REQ_31.9.4.4.2.1	The <**rollupCondition**> element shall contain 0 or 1 **operator** attribute.
REQ_31.9.4.4.2.1.1	The **operator** attribute's value shall be one of the following restricted tokens: • not • noOp (default if the attribute is not provided)
REQ_31.9.4.4.2.2	The <**rollupCondition**> element shall contain 1 and only 1 **condition** attribute.
REQ_31.9.4.4.2.2.1	The **condition** attribute's value shall be one of the following restricted tokens: • satisfied • objectiveStatusKnown • objectiveMeasureKnown • completed • activityProgressKnown

REQ ID	Requirement
	• `attempted` • `attemptLimitExceeded` • `timeLimitExceeded` • `outsideAvailableTimeRange` **ADL NOTE:** If no **condition** attribute is provided the **action** defined for the condition is always taken.
REQ_31.9.4.5	The <**rollupRule**> element shall contain 1 and only 1 <**rollupAction**> child elements.
REQ_31.9.4.5.1	The <**rollupAction**> element shall contain 1 and only 1 **action** attribute.
REQ_31.9.4.5.1.1	The **action** attribute's value shall be one of the following restricted tokens: • `satisfied` notSatisfied • `completed` • `incomplete`
REQ_31.10	The <**sequencing**> element shall contain 0 or 1 <**objectives**> child element.
REQ_31.10.1	The <**objectives**> element shall contain 1 and only 1 <**primaryObjective**> child element.
REQ_31.10.1.1	The <**primaryObjective**> element shall contain 0 or 1 **satisfiedByMeasure** attribute.
REQ_31.10.1.1.1	The **satisfiedByMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.1.1.2	The **satisfiedByMeasure** attribute's value shall be either: • `true`, or • `false` (default if no **satisfiedByMeasure** attribute is present).
REQ_31.10.1.2	The <**primaryObjective**> element shall contain 0 or 1 **objectiveID** attribute.
REQ_31.10.1.2.1	The **objectiveID** attribute's value shall be represented as an xs:anyURI type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_31.10.1.2.2	If a <**primaryObjective**> element contains an objective map (i.e., <**mapInfo**> element), then the **objectiveID** attribute is required.
REQ_31.10.1.2.3	For a given set of objectives defined for an activity (i.e., one <**primaryObjective**> and multiple <**objective**> elements), the **objectiveID** attributes values for all of the objective IDs shall be unique.
REQ_31.10.1.2.4	Since the underlying data type of a **objectiveID** is a unique identifier, then the value held by the **objectiveID** shall not be an empty characterstring nor contain all white space characters.
REQ_31.10.1.3	The <**primaryObjective**> element shall contain 0 or 1 <**minNormalizedMeasure**> child element.

REQ ID	Requirement
REQ_31.10.1.3.1	The **minNormalizedMeasure** element's value shall be a valid xs:decimal in the range of -1.0000 and 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal.
REQ_31.10.1.3.2	If no <**minNormalizedMeasure**> is provided the default value of 1.0 shall be used to represent the minimum normalized measure.
REQ_31.10.1.4	The <**primaryObjective**> element shall contain 0 or More <**mapInfo**> child elements.
REQ_31.10.1.4.1	The <**mapInfo**> element shall contain 1 and only 1 **targetObjectiveID** attribute.
REQ_31.10.1.4.1.1	The **targetObjectiveID** attribute's value shall be represented as an xs:anyURI type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_31.10.1.4.1.2	Since the underlying data type of a **targetObjectiveID** is a unique identifier, then the value held by the **targetObjectiveID** shall not be an empty characterstring nor contain all white space characters.
REQ_31.10.1.4.2	The <**mapInfo**> element shall contain 0 or 1 **readSatisfiedStatus** attribute.
REQ_31.10.1.4.2.1	The **readSatisfiedStatus** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.1.4.2.2	The **readSatisfiedStatus** attribute's value shall be either: • `true` (default if no **readSatisfiedStatus** attribute is present), or • `false`
REQ_31.10.1.4.3	The <**mapInfo**> element shall contain 0 or 1 **readNormalizedMeasure** attribute.
REQ_31.10.1.4.3.1	The **readNormalizedMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.1.4.3.2	The **readNormalizedMeasure** attribute's value shall be either: • `true` (default if no **readNormalizedMeasure** attribute is present), or • `false`
REQ_31.10.1.4.4	The <**mapInfo**> element shall contain 0 or 1 **writeSatisfiedStatus** attribute.
REQ_31.10.1.4.4.1	The **writeSatisfiedStatus** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.1.4.4.2	The **writeSatisfiedStatus** attribute's value shall be either: • `true`, or • `false` (default if no **writeSatisfiedStatus** attribute is present) .
REQ_31.10.1.4.5	The <**mapInfo**> element shall contain 0 or 1 **writeNormalizedMeasure** attribute.

REQ ID	Requirement
REQ_31.10.1.4.5.1	The **writeNormalizedMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.1.4.5.2	The **writeNormalizedMeasure** attribute's value shall be either: • `true`, or • `false` (default if no **writeNormalizedMeasure** attribute is present) .
REQ_31.10.1.4.6	The <**mapInfo**> element shall adhere to the following when dealing with Read Objective Maps:
REQ_31.10.1.4.6.1	If multiple <**mapInfo**> elements exist for an objective (<**primaryObjective**> or <**objective**>) then only one <**mapInfo**> element shall have the **readSatisfiedStatus** attribute set to `true`.
REQ_31.10.1.4.6.2	If multiple <**mapInfo**> elements exist for an objective (<**primaryObjective**> or <**objective**>) then only one <**mapInfo**> element shall have the **readNormalizedMeasure** attribute set to `true`.
REQ_31.10.1.4.7	The <**mapInfo**> element shall adhere to the following when dealing with Write Objective Maps:
REQ_31.10.1.4.7.1	For an activity, if multiple objectives (<**primaryObjective**> or <**objective**>) have <**mapInfo**> elements that share the same **targetObjectiveID**, then only one of the objectives shall have the **writeSatisfiedStatus** attribute set to `true`.
REQ_31.10.1.4.7.2	For an activity, if multiple objectives (<**primaryObjective**> or <**objective**>) have <**mapInfo**> elements that share the same **targetObjectiveID**, then only one of the objectives shall have the **writeNormalizedMeasure** attribute set to `true`.
REQ_31.10.2	The <**objectives**> element shall contain 0 or More <**objective**> child element.
REQ_31.10.2.1	The <**objective**> element shall contain 0 or 1 **satisfiedByMeasure** attribute.
REQ_31.10.2.1.1	The **satisfiedByMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.2.1.2	The **satisfiedByMeasure** attribute's value shall be either: • `true`, or • `false` (default if no **satisfiedByMeasure** attribute is present).
REQ_31.10.2.2	The <**objective**> element shall contain 1 and only 1 **objectiveID** attribute.
REQ_31.10.2.2.1	The **objectiveID** attribute's value shall be represented as an xs:anyURI type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_31.10.2.2.2	For a given set of objectives defined for an activity (i.e., one <**primaryObjective**> and multiple <**objective**> elements), the **objectiveID** attributes values for all of the objective IDs shall be unique.
REQ_31.10.2.2.3	Since the underlying data type of a **objectiveID** is a unique identifier, then the value held by the **objectiveID** shall not be an empty characterstring nor contain all white space characters.

REQ ID	Requirement
REQ_31.10.2.3	The <**objective**> element shall contain 0 or 1 <**minNormalizedMeasure**> child element.
REQ_31.10.2.3.1	The **minNormalizedMeasure** element's value shall be a valid xs:decimal in the range of -1.0000 and 1.0000 (precision to at least 4 significant decimal places). Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:decimal.
REQ_31.10.2.3.2	If no <**minNormalizedMeasure**> is provided the default value of 1.0 shall be used to represent the minimum normalized measure.
REQ_31.10.2.4	The <**objective**> element shall contain 0 or More <**mapInfo**> child elements.
REQ_31.10.2.4.1	The <**mapInfo**> element shall contain 1 and only 1 **targetObjectiveID** attribute.
REQ_31.10.2.4.1.1	The **targetObjectiveID** attribute's value shall be represented as an xs:anyURI type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_31.10.2.4.2	The <**mapInfo**> element shall contain 0 or 1 **readSatisfiedStatus** attribute.
REQ_31.10.2.4.2.1	The **readSatisfiedStatus** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.2.4.2.2	The **readSatisfiedStatus** attribute's value shall be either: • `true` (default if no **readSatisfiedStatus** attribute is present), or • `false`
REQ_31.10.2.4.3	The <**mapInfo**> element shall contain 0 or 1 **readNormalizedMeasure** attribute.
REQ_31.10.2.4.3.1	The **readNormalizedMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.2.4.3.2	The **readNormalizedMeasure** attribute's value shall be either: • `true` (default if no **readNormalizedMeasure** attribute is present), or • `false`
REQ_31.10.2.4.4	The <**mapInfo**> element shall contain 0 or 1 **writeSatisfiedStatus** attribute.
REQ_31.10.2.4.4.1	The **writeSatisfiedStatus** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.10.2.4.4.2	The **writeSatisfiedStatus** attribute's value shall be either: • `true`, or • `false` (default if no **writeSatisfiedStatus** attribute is present).
REQ_31.10.2.4.5	The <**mapInfo**> element shall contain 0 or 1 **writeNormalizedMeasure** attribute.
REQ_31.10.2.4.5.1	The **writeNormalizedMeasure** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of

REQ ID	Requirement
	values of an xs:boolean.
REQ_31.10.2.4.5.2	The **writeNormalizedMeasure** attribute's value shall be either: • `true`, or • `false` (default if no **writeNormalizedMeasure** attribute is present).
REQ_31.11	The <**sequencing**> element shall contain 0 or 1 <**randomizationControls**> child element.
REQ_31.11.1	The <**randomizationControls**> element shall contain 0 or 1 **randomizationTiming** attribute.
REQ_31.11.1.1	The **randomizationTiming** attribute's value shall be either: • `once` • `onEachNewAttempt` • `never`
REQ_31.11.1.2	If no **randomizationTiming** attribute is defined then randomization on the set of activities shall never be performed by the LMS.
REQ_31.11.2	The <**randomizationControls**> element shall contain 0 or 1 **selectCount** attribute.
REQ_31.11.2.1	The **selectCount** attribute's value shall be a non-negative integer.
REQ_31.11.2.2	If no **selectCount** attribute is defined for the <**randomizationControls**> element then the default value of 0 shall be used.
REQ_31.11.3	The <**randomizationControls**> element shall contain 0 or 1 **reorderChildren** attribute.
REQ_31.11.3.1	The **reorderChildren** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean
REQ_31.11.3.2	The **reorderChildren** attribute's value shall be either: • `true`, or • `false` (default if no **reorderChildren** attribute is present).
REQ_31.11.4	The <**randomizationControls**> element shall contain 0 or 1 **selectionTiming** attribute.
REQ_31.11.4.1	The **selectionTiming** attribute's value shall be either: • `once` • `onEachNewAttempt` • `never`
REQ_31.11.4.2	If no **selectionTiming** attribute is defined then selection from the set of activities shall never be performed by the LMS.
REQ_31.12	The <**sequencing**> element shall contain 0 or 1 <**deliveryControls**> child element.
REQ_31.12.1	The <**deliveryControls**> element shall contain 0 or 1 **tracked** attribute.
REQ_31.12.1.1	The **tracked** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.

REQ ID	Requirement
REQ_31.12.1.2	The **tracked** attribute's value shall be either: • `true` (default if no **tracked** attribute is present), or • `false`
REQ_31.12.2	The <**deliveryControls**> element shall contain 0 or 1 **completionSetByContent** attribute.
REQ_31.12.2.1	The **completionSetByContent** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.12.2.2	The **completionSetByContent** attribute's value shall be either: • `true`, or • `false` (default if no **completionSetByContent** attribute is present).
REQ_31.12.3	The <**deliveryControls**> element shall contain 0 or 1 **objectiveSetByContent** attribute.
REQ_31.12.3.1	The **objectiveSetByContent** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.12.3.2	The **objectiveSetByContent** attribute's value shall be either • `true`, or • `false` (default if no **objectiveSetByContent** attribute is present)
REQ_31.13	The <**sequencing**> element shall contain 0 or 1 <**adlseq:constrainedChoiceConsiderations**> child element.
REQ_31.13.1	The <**adlseq:constrainedChoiceConsiderations**> element shall contain 0 or 1 **preventActivation** attribute.
REQ_31.13.1.1	The **preventActivation** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.13.1.2	The **preventActivation** attribute's value shall be either: • `true`, or • `false` (default if no **preventActivation** attribute is present).
REQ_31.13.2	The <**adlseq:constrainedChoiceConsiderations**> element shall contain 0 or 1 **constrainChoice** attribute.
REQ_31.13.2.1	The **constrainChoice** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.13.2.2	The **constrainChoice** attribute's value shall be either: • `true`, or • `false` (default if no **constrainChoice** attribute is present).

REQ ID	Requirement
REQ_31.14	The **<sequencing>** element shall contain 0 or 1 **<adlseq:rollupConsiderations>** child element.
REQ_31.14.1	The **<adlseq:rollupConsiderations>** element shall contain 0 or 1 **requiredForSatisfied** attribute.
REQ_31.14.1.1	The **requiredForSatisfied** attribute's value shall be represented as an xs:token type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:token.
REQ_31.14.1.2	The **requiredForSatisfied** attribute's value shall be one of the following tokens: • `ifAttempted` • `ifNotSkipped` • `ifNotSuspended` • `always` **ADL NOTE:** If the **requiredForSatisfied** attribute is not provided then activity is always used in rollup rule processing.
REQ_31.14.2	The **<adlseq:rollupConsiderations>** element shall contain 0 or 1 **requiredForNotSatisfied** attribute.
REQ_31.14.2.1	The **requiredForNotSatisfied** attribute's value shall be represented as an xs:token type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:token.
REQ_31.14.2.2	The **requiredForNotSatisfied** attribute's value shall be one of the following restricted tokens: • `ifAttempted` • `ifNotSkipped` • `ifNotSuspended` • `always` **ADL NOTE**: If the requiredForNotSatisfied attribute is not provided then activity is always used in rollup rule processing.
REQ_31.14.3	The **<adlseq:rollupConsiderations>** element shall contain 0 or 1 **requiredForCompleted** attribute.
REQ_31.14.3.1	The **requiredForCompleted** attribute's value shall be represented as an xs:token type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:token.
REQ_31.14.3.2	The **requiredForCompleted** attribute's value shall be one of the following restricted tokens: • `ifAttempted` • `ifNotSkipped` • `ifNotSuspended` • `always` **ADL NOTE**: If the **requiredForCompleted** attribute is not provided then activity is always used in rollup rule processing.

REQ ID	Requirement
REQ_31.14.4	The <**adlseq:rollupConsiderations**> element shall contain 0 or 1 **requiredForIncomplete** attribute.
REQ_31.14.4.1	The **requiredForIncomplete** attribute's value shall be represented as an xs:token type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:token.
REQ_31.14.4.2	The **requiredForIncomplete** attribute's value shall be either • `ifAttempted` • `ifNotSkipped` • `ifNotSuspended` • `always` **ADL NOTE:** If the **requiredForIncomplete** attribute is not provided then activity is always used in rollup rule processing.
REQ_31.14.5	The <**adlseq:rollupConsiderations**> element shall contain 0 or 1 **measureSatisfactionIfActive** attribute.
REQ_31.14.5.1	The **measureSatisfcationIfActive** attribute's value shall be represented as an xs:boolean type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:boolean.
REQ_31.14.5.2	The **measureSatisfactionIfActive** attribute's value shall be either: • `true` (default if no **measureSatisfactionIfActive** attribute is present), or • `false`
REQ_32	The <**manifest**> element shall contain 0 or 1 <**sequencingCollection**> element.
REQ_32.1	The <**sequencingCollection**> element shall contain 1 or More <**sequencing**> element.
REQ_32.1.1	The <**sequencing**> element shall contain 1 and only 1 **ID** attribute.
REQ_32.1.1.1	The **ID** attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_32.1.1.2	The **ID** attribute shall be unique within the Manifest.
REQ_32.1.2	The <**sequencing**> element shall not have a defined **IDRef**.
REQ_32.1.3	The XML instance shall adhere to all other requirements defined for the <**sequencing**> element (refer to requirement REQ_31)

3.1.4. Navigation and Presentation Extensions Conformance Requirements

The SCORM SN defines the requirements that must be met when encoding specific navigation strategies in a Content Aggregation Package Application Profile Manifest. The elements listed in Table 3.1.4a are optional. If a content package manifest uses the navigation elements defined in Table 3.1.4a, the manifest shall conform to the requirements defined below.

Table 3.1.4a: Navigation and Presentation Extensions Conformance Requirements

REQ ID	Requirement
REQ_33	The **<item>** element shall contain 0 or 1 **<adlnav:presentation>** element. The element shall only be used with **<item>** elements that reference SCO or Asset resources.
REQ_33.1	The **<adlnav:presentation>** element shall contain 0 or 1 **<adlnav:navigationInterface>** element.
REQ_33.1.1	The **<adlnav:navigationInterface>** element shall contain 0 or More **<adlnav:hideLMSUI>** element.
REQ_33.1.1.1	The **<adlnav:hideLMSUI>** element's value shall be one of the following restricted vocabulary tokens: • `previous` • `continue` • `exit` • `exitAll` • `abandon` • `abandonAll` • `suspendAll`

3.1.5. Resource Package Manifest Conformance Requirements

The SCORM Resource Package Application Profile defines a specification for packaging learning resources (for example, Assets and SCOs) without having to provide a specific organization, learning context or curricular taxonomy. Packaging learning resources provides a common medium for exchange.

The SCORM Resource Application Profile Package Manifest shall adhere to the requirements defined in Table 3.1.5a to be considered conformant to the CP CAM 1.0 and CP RTE 1.0 conformance categories.

Table 3.1.5a: Resource Package Manifest Conformance Requirements

REQ ID	Requirement
REQ_29	The Resource Application Profile Package's Manifest XML file (`imsmanifest.xml`) shall contain 1 and only 1 root **<manifest>** element.
REQ_29.1	The **<manifest>** element shall contain 1 and only 1 **identifier** attribute.
REQ_29.1.1	The **identifier** attribute's value shall be represented as an xs:ID type. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID
REQ_29.1.2	The **identifier** attribute shall be unique within the Manifest.
REQ_29.2	The **<manifest>** element shall contain 0 or 1 **version** attribute.
REQ_29.2.1	The **version** attribute shall be represented as a xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_29.3	The **<manifest>** element shall contain 0 or 1 **xml:base** attribute.
REQ_29.3.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI
REQ_29.3.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_29.3.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_29.3.4	The **xml:base** attribute's value shall end with a trailing slash ("/").
REQ_29.4	The **<manifest>** element shall contain 1 and only 1 **<metadata>** child element.
REQ_29.4.1	The child elements of an **<metadata>** element shall exist in the specified order: • **<schema>** • **<schemaversion>** • **{Metadata}** The order is not defined for extension elements, they are only required to be placed after the **<schemaversion>** element.

REQ ID	Requirement
REQ_29.4.2	The <**metadata**> element shall contain 1 and only 1 <**schema**> child element.
REQ_29.4.2.1	The <**schema**> element's value shall be set to the restricted characterstring token: • `ADL SCORM`
REQ_29.4.3	The <**metadata**> element shall contain 1 and only 1 <**schemaversion**> child element.
REQ_29.4.3.1	The <**schemaversion**> element's value shall be set to the restricted characterstring token: • `2004 3rd Edition`
REQ_29.4.4	The <**metadata**> element shall be the container for 0 or More **{Metadata}**. **ADL NOTE: {Metadata}** can be represented in two ways, either inline extensions to the content package manifest or using the <**adlcp:location**> element to reference a stand-alone XML instance document.
REQ_29.4.4.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_29.4.4.1.1	The <**metadata**> element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the content aggregation.
REQ_29.4.4.1.2	The <**metadata**> element shall contain 0 or More <**adlcp:location**> child elements to define the location (URL) of the **{Metadata}** describing the content aggregation. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_29.4.4.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_29.5	The <**manifest**> element shall contain 1 and only 1 <**organizations**> element.
REQ_29.5.1	The <**organizations**> element shall be an empty XML element (i.e., <**organizations**/>. **ADL NOTE:** An empty element, by definition, is permitted to contain attributes.
REQ_29.6	The <**manifest**> element shall contain 1 and only 1 <**resources**> child element.
REQ_29.6.1	The <**resources**> element shall contain 0 or 1 **xml:base** attribute
REQ_29.6.1.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI
REQ_29.6.1.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_29.6.1.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_29.6.1.4	The **xml:base** attribute's value shall end with a trailing slash ("/").

REQ ID	Requirement
REQ_29.6.2	The child elements of an **<resources>** element shall exist in the specified order: • **<resource>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<resource>** element.
REQ_29.6.3	The **<resources>** element shall contain 0 or More **<resource>** elements.
REQ_29.6.3.1	The **<resource>** element shall contain 1 and only 1 **identifier** attribute.
REQ_29.6.3.1.1	The **identifier** attribute's value shall be a valid xs:ID. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:ID.
REQ_29.6.3.1.2	The **identifier** attribute's value shall be unique within the Manifest.
REQ_29.6.3.2	The **<resource>** element shall contain 1 and only 1 **type** attribute
REQ_29.6.3.2.1	The **type** attribute's value shall be a valid xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string.
REQ_29.6.3.3	The **<resource>** element shall contain 0 or 1 **href** attribute
REQ_29.6.3.3.1	The **href** attribute shall be a characterstring that represents the URL (as defined in IETF RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax) for the resource. **ADL NOTE:** The **href** attribute is affected by the **xml:base** attribute if provided.
REQ_29.6.3.3.2	The **href** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_29.6.3.3.3	The **href** attribute's value shall not begin with a leading slash ("/").
REQ_29.6.3.4	The **<resource>** element shall contain 1 and only 1 **adlcp:scormType** attribute
REQ_29.6.3.4.1	The **adlcp:scormType** value shall be a characterstring where the characterstring is either: • `sco` if the resource is a SCO, or • `asset` if the resource is an Asset.
REQ_29.6.3.5	The **<resource>** element shall contain 0 or 1 **xml:base** attribute
REQ_29.6.3.5.1	The **xml:base** attribute's value shall be represented as an xs:anyURI Datatype. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:anyURI.
REQ_29.6.3.5.2	The **xml:base** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_29.6.3.5.3	The **xml:base** attribute's value shall not begin with a leading slash ("/").
REQ_29.6.3.5.4	The **xml:base** attribute's value shall end with a trailing slash ("/").

REQ ID	Requirement
REQ_29.6.3.7	The child elements of an **<resource>** element shall exist in the specified order: • **<metadata>** • **<file>** • **<dependency>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<dependency>** element.
REQ_29.6.3.8	The **<resource>** element shall contain 0 or 1 **<metadata>** child element.
REQ_29.6.3.8.1	The **<metadata>** element shall be the container for 0 or More **{Metadata}**.
REQ_29.6.3.8.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_29.6.3.8.1.1.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the Asset resource.
REQ_29.6.3.8.1.1.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the Asset resource. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_29.6.3.8.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_29.6.3.8.1.2	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_29.6.3.8.1.2.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the SCO resource.
REQ_29.6.3.8.1.2.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the SCO resource. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_29.6.3.8.1.2.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_29.6.3.9	The **<resource>** element shall contain 0 or More **<file>** child elements.
REQ_29.6.3.9.1	The **<file>** element shall identify 0 or More local files that this resource is dependent on. **ADL NOTE:** For all files that are required for delivery and are local to the content package (physically located in a content package), a **<file>** element shall be used to represent the file relative to the resource in which it is used.
REQ_29.6.3.9.1.1	If the **<resource>** is local to the content package, then a **<file>** element is required that represents the **<resource>** itself. The **href** attribute of the **<file>**

REQ ID	Requirement
	element shall be identical to the **href** attribute of the **<resource>**.
REQ_29.6.3.9.2	The **<file>** element shall contain 1 and only 1 **href** attribute.
REQ_29.6.3.9.2.1	The **href** attribute shall be a characterstring that represents the URL (as defined in IETF RFC 2395 – Uniform Resource Identifiers (URI): Generic Syntax) for the resource. **ADL NOTE:** The **href** attribute is affected by the **xml:base** attribute if provided.
REQ_29.6.3.9.2.2	The **href** attribute's value shall not contain any backward slashes ("\"). **ADL NOTE**: If a backward slash is needed, then the value shall be properly encoded.
REQ_29.6.3.9.2.3	The **href** attribute's value shall not begin with a leading slash ("/").
REQ_29.6.3.9.3	The child elements of an **<file>** element shall exist in the specified order: • **<metadata>** • **extension elements** The order is not defined for extension elements, they are only required to be placed after the **<metadata>** element.
REQ_29.6.3.9.4	The **<file>** element shall contain 0 or 1 **<metadata>** child element.
REQ_29.6.3.9.4.1	The **<metadata>** element shall be the container for 0 or More **{Metadata}**.
REQ_29.6.3.9.4.1.1	If providing Metadata, the **{Metadata}** shall be well-formed and valid according to the cooresponding Controlling Document (e.g., XSD).
REQ_29.6.3.9.4.1.1.1	The **<metadata>** element shall contain 0 or More XML extensions to the IMS Content Packaging namespace. These inline extensions shall define the **{Metadata}** describing the Asset.
REQ_29.6.3.9.4.1.1.2	The **<metadata>** element shall contain 0 or More **<adlcp:location>** child elements to define the location (URL) of the **{Metadata}** describing the Asset. **ADL NOTE:** The value of the URL is affected by the **xml:base** attribute if provided.
REQ_29.6.3.9.4.1.1.2.1	The value of this element shall be a characterstring that represents the URL (as defined in IETF RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax).
REQ_29.6.3.10	The **<resource>** element shall contain 0 or More **<dependency>** child elements.
REQ_29.6.3.10.1	The **<dependency>** element shall contain 1 and only 1 **identifierref** attribute.
REQ_29.6.3.10.1.1	The **identifierref** attribute's value shall be an xs:string. Refer to W3C XML Schema Part 2: Datatype for information on the valid set of values of an xs:string
REQ_29.6.3.10.2	The **<dependency>** element shall be an XML empty element (i.e., **<dependency/>**. **ADL NOTE:** An empty element, by definition, is permitted to contain attributes.
REQ_29.7	The **<manifest>** element (root) shall contain 0 or More **<manifest>** child elements, i.e., (sub)manifests. **ADL Note:** The IMS Global Consortium, Inc., is working on a new version of

REQ ID	Requirement
	the IMS Content Packaging Specification. One of the major issues that IMS is resolving deals with (sub)manifests, their use, requirements of use and XML syntax requirements. At this time, ADL recommends not to use (sub)manifests until completion of the IMS work. Any questions, concerns or further recommendations on (sub)manifests should be sent to ADL

This page intentionally left blank.

SECTION 4
SCO Conformance Requirements

This page intentionally left blank.

4.1. SCO Conformance Requirements

This section describes the detailed requirements that must be implemented by a SCO to be SCORM 2004 3rd Edition conformant. A SCO is required to support the various aspects defined in the SCORM Run-Time Environment [1].

The conformance requirements for a SCO are divided into the following sections:

- Section 4.1.1: Launch Conformance Requirements
- Section 4.1.2: API Conformance Requirements
- Section 4.1.3: Run-Time Environment Data Model Conformance Requirements
- Section 4.1.4: Run-Time Environment Data Model Type Conformance Requirements
- Section 4.1.5: Run-Time Navigation Data Model Conformance Requirements

The purpose of the SCO Conformance Test is to verify that SCO implements the conformance requirements as outlined in this section. The SCO Conformance Test is designed to test the following conformance categories:

- Conformance Category 1: **SCO RTE 1.0** – The SCO is conformant to the requirements defined in the SCORM 2004 3rd Edition Run-Time Environment Version 1.0.

The SCO Conformance Test verifies that a SCO implements the conformance requirements as outlined in this section. The SCO Conformance Test software simulates a "SCORM 2004 3rd Edition conformant LMS." The SCO is launched by the test software and is expected to search for and find the LMS' API Instance. The test software then services and audits any API method call that is made by the SCO. As the test suite operator executes the SCO's functionality (simulates the learner experience), the test software audits the SCORM Run-Time Environment Data Model and SCORM Run-Time Navigation Data Model elements that are implemented by the SCO.

The conformance category and conformance label assigned to the SCO is dependent upon the SCO's ability to adhere to the conformance requirements. A SCO at a minimum shall:

- Search for and find an LMS provided API Instance
- Invoke Initialize ("") to initialize communication
- Invoke Terminate("") to terminate communication

The rest of the SCORM API methods are optional for a SCO to implement and use. If a SCO attempts to use any API method, then the SCO Conformance Test will determine if the SCO uses the API method correctly. If the SCO attempts to retrieve (GetValue) or store (SetValue) any SCORM Run-Time Environment Data Model or SCORM Run-Time Navigation Data Model element, then the SCO Conformance Test will determine if the

SCO uses the data model element correctly. Any failure with the optional API methods or usage of the data models will cause the SCO to fail the test.

It is important to recognize that certain SCOs may be very simple in nature, offering a single logical path of execution from start to finish. Other SCOs may be more complex, having several possible paths of execution that are conditional, based on user performance or user personalization variables, for example. For this reason, it is not possible to programmatically test all conditional branches of a SCO using a "black box" testing approach. Furthermore, without placing specific constraints on implementation technologies, it is not feasible to inspect the implementation details of the SCO to validate conformance, as would be required using a "white box" testing approach.

ADL has chosen a subjective approach that involves a compromise between the "black box" and "white box" approaches. When testing a SCO that contains conditional logic that involves the potential for the SCO to interact with an LMS using data model element sets that vary across the conditional paths, or to execute varying sets of API methods, the path for the test is left to the discretion of the test suite operator. The test suite operator may, if testing multiple SCOs from a single organization, elect to employ different strategies across multiple SCOs, or even within a single SCO.

For this reason, the SCO Conformance Test does not guarantee that the SCO correctly implements the SCORM 2004 3rd Edition Run-Time Environment in all cases, but only within the bounds of the scenario that was used for an instance of testing.

It is also important to realize that the SCO Conformance Test does not guarantee that the SCO is without defects. The test software does not validate that all aspects of the SCO implementation are accurate and/or correct. For example, it is quite possible that a SCO can be tested and passed as SCORM 2004 3rd Edition conformant within one of the previously mentioned categories and still contain run-time defects (e.g. broken links, JavaScript run-time errors, etc.), and/or be instructionally unsound.

4.1.1. Launch Conformance Requirements

Once launched by an LMS, the SCO is responsible for finding the LMS provided API Instance and correctly issuing the following API method calls:

- Initialize ("") to indicate that the SCO is ready to communicate with the LMS.
- Terminate("") to indicate that the SCO has finished communicating with the LMS.

If a SCO makes use of any of the other API method calls, then the SCO shall invoke those API method calls in accordance with the conformance requirements for those API methods.

Table 4.1.1a: SCO Launch Conformance Requirements

REQ ID	Requirement

REQ ID	Requirement
REQ_26	When launched by a known conformant LMS, the SCO shall search for and find the LMS API Adapter DOM object named `API_1484_11` by searching the parent and opener DOM window hierarchy.
REQ_26.1	The SCO shall locate an instance of the API implementation in any of the following locations, in the specified order of precedence and stop as soon as an instance is found.
REQ_26.1.1	The SCO shall search for the API instance in any window in the chain of parents of the current window, if any exist, until the top window of the parent chain is reached.
REQ_26.1.2	The SCO shall search for the API instance in an opener window, if an opener exists.
REQ_26.1.3	The SCO shall search for the API instance in any window in the chain of parents of the opener window, if an opener exists, until the top window of the opener's parent chain is reached.
REQ_27	The SCO shall be implemented such that it does not require that it be the top-level window in the DOM window hierarchy upon launch. The SCO must not contain DOM documents that reference relative documents within the SCO using the window.top DOM object.

4.1.2. API Conformance Requirements

All SCOs, by definition, communicate with an LMS. Once the SCO has located the LMS provided API Instance, the SCO must invoke at a minimum the Initialize("") API method call to initialize communication with an LMS. Once the SCO has determined that the communication session is no longer needed, the SCO shall invoke the Terminate("") API method call. If the SCO must invoke any of the other API method calls, the SCO is responsible for ensuring conformance to the requirements of those API requirements.

The following conformance requirements are written with the assumption that the SCO is invoking all of the API method calls.

The SCO shall adhere to the requirements defined in Table 4.1.2a to be considered conformant to the SCO RTE 1.0 conformance category.

Table 4.1.2a: SCO API Conformance Requirements

REQ ID	Requirement
REQ_12	The SCO shall invoke the **Initialize()** API method in accordance with the following requirements:
REQ_12.1	The SCO shall invoke the **Initialize()** API method upon launch of the SCO and before invoking any other API methods, except for **GetLastError()**, **GetErrorString()** and/or **GetDiagnostic()**.
REQ_12.2	The SCO shall invoke the **Initialize()** API method with a single parameter that contains an empty characterstring ("").
REQ_12.3	The SCO shall accept a characterstring return value when invoking the **Initialize()** API method.

REQ ID	Requirement
REQ_13	The SCO shall invoke the **Terminate()** API method in accordance with the following requirements:
REQ_13.1	The SCO shall invoke the **Terminate()** API method when it is finished communicating with the LMS.
REQ_13.2	The SCO shall invoke the **Terminate()** API method with a single parameter that contains an empty characterstring("").
REQ_13.3	The SCO shall accept a characterstring return value when invoking the **Terminate()** API method.
REQ_13.4	The SCO shall not invoke any other API methods after a successful call to **Terminate()**, except for **Terminate()**, **GetLastError()**, **GetErrorString()** and/or **GetDiagnostic()**.
REQ_14	The SCO shall invoke the **SetValue()** API method in accordance with the following requirements:
REQ_14.1	The SCO shall invoke the **SetValue()** API method to set (writeable) data model values in an LMS.
REQ_14.2	The SCO shall invoke the **SetValue()** API method with two parameters.
REQ_14.2.1	The first parameter to the **SetValue()** API method shall be a fully qualified, case sensitive characterstring containing the name of the data model element requested to be set by the SCO.
REQ_14.2.2	The second parameter to the **SetValue()** API method shall be the characterstring containing the value for the data model element that the SCO is requesting to be set.
REQ_14.2.3	The second parameter to the **SetValue()** API method shall be conformant to the requirements defined by the parameter 1 (data model element to be set). See the SCORM Data Model Conformance Requirements for details.
REQ_14.3	The SCO shall accept a characterstring return value when invoking the **SetValue()** API method.
REQ_15	The SCO shall invoke the **GetValue()** API method in accordance with the following requirements:
REQ_15.1	The SCO shall use the **GetValue()** API method to get (readable) data model element values from the LMS.
REQ_15.2	The SCO shall invoke the **GetValue()** API method with one parameter.
REQ_15.2.1	The parameter passed into the **GetValue()** API method shall be a fully qualified, case sensitive name of the data model element requested to be retrieved (read) by the SCO.
REQ_15.3	The SCO shall accept a characterstring return value when invoking the **GetValue()** API method.
REQ_16	The SCO shall invoke the **GetLastError()** API method in accordance with the following requirements:
REQ_16.1	The SCO shall invoke the **GetLastError()** API method with no parameters.
REQ_16.2	The SCO shall accept a characterstring return value when invoking the GetLastError API method.
REQ_17	The SCO shall invoke the **GetErrorString()** API method in accordance with the following requirements:
REQ_17.1	The SCO shall invoke the **GetErrorString()** API method with one characterstring parameter that contains a valid API Error Code.

REQ ID	Requirement
REQ_17.2	The SCO shall accept a characterstring return value when invoking the **GetErrorString()** API method.
REQ_18	The SCO shall invoke the **GetDiagnostic()** API method in accordance with the following requirements:
REQ_18.1	The SCO shall invoke the **GetDiagnostic()** API method with one characterstring parameter.
REQ_18.2	The SCO shall accept a characterstring return when invoking the **GetDiagnostic()** API method.
REQ_19	The SCO shall invoke the **Commit()** API method in accordance with the following requirements:
REQ_19.1	The SCO shall invoke the **Commit()** API method with a single characterstring parameter that contains an empty characterstring ("").
REQ_19.2	The SCO shall accept a characterstring return value when invoking the **Commit()** API method.
REQ_20	SCOs shall utilize the API through an ECMAScript (JavaScript) binding:
REQ_20.1	The SCO shall invoke the API implementation methods using ECMAScript calling conventions.
REQ_20.2	The SCO shall implement all API implementation method parameters as ECMAScript characterstrings.
REQ_20.3	The SCO shall accept all API implementation method return values as ECMAScript characterstrings.
REQ_20.4	The SCO shall invoke all API methods, by calling them by their case-sensitive names.
REQ_20.5	The SCO shall encode method parameters and return values that represent integers, real numbers, durations and times as they would be by the ECMAScript-to-string cast conversion.

4.1.3. Run-Time Environment Data Model Requirements

An LMS is required to implement all SCORM Run-Time Environment Data Model elements to be considered SCORM 2004 3rd Edition conformant.

A SCO is not required to exchange data with an LMS in order to be SCORM 2004 3rd Edition conformant. Additionally, a SCO may attempt to exchange data with an LMS using elements that are not part of the SCORM Run-Time Environment Data Model. Doing so is not recommended because interoperability is hindered; however, this will not prevent the SCO from being SCORM 2004 3rd Edition conformant.

If the SCO does implement the ability to exchange data with an LMS using the SCORM Run-Time Environment Data Model, it must use the defined elements, within the bounds of the requirements as defined in Table 4.1.3a.

The SCO shall adhere to the requirements defined in Table 4.1.3a to be considered conformant to the SCO RTE 1.0 conformance category.

Table 4.1.3a: SCO Run-Time Environment Data Model Requirements

REQ ID	Requirement
REQ_56	The SCO shall only invoke a GetValue() request for the **cmi._version** data model element to determine the data model version supported.
REQ_93	If utilizing the **cmi.comments_from_learner**, the SCO shall adhere to the following requirements.
REQ_93.1	The SCO shall only invoke a GetValue() request for the **cmi.comments_from_learner._children** data model element.
REQ_93.2	The SCO shall only invoke a GetValue() request for the **cmi.comments_from_learner._count** data model element.
REQ_93.3	If utilizing the **cmi.comments_from_learner.n.comment** the SCO shall adhere to the following requirements.
REQ_93.3.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.comments_from_learner.n.comment** data model element.
REQ_93.3.2	During a SetValue() request, the SCO shall ensure that the **cmi.comments_from_learner.n.comment** value is a valid a localized_string_type.
REQ_93.4	If utilizing the **cmi.comments_from_learner.n.location** the SCO shall adhere to the following requirements.
REQ_93.4.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.comments_from_learner.n.location** data model element.
REQ_93.4.2	During a SetValue() request, the SCO shall ensure that the **cmi.comments_from_learner.n.location** value is a valid characterstring.
REQ_93.5	If utilizing the **cmi.comments_from_learner.n.timestamp** the SCO shall adhere to the following requirements.

REQ ID	Requirement
REQ_93.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.comments_from_learner.n.timestamp** data model element.
REQ_93.5.2	During a SetValue() request, the SCO shall ensure that the **cmi.comments_from_learner.n.timestamp** value is a valid time (second,10,0).
REQ_93.6	During a SetValue() request, the SCO shall ensure that the index value (n) is either less than or equal to the current number of comments being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. Arrays are treated as packed arrays and have to be set in order.
REQ_93.7	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of comments being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_94	If utilizing the **cmi.comments_from_lms** the SCO shall adhere to the following requirements.
REQ_94.1	The SCO shall only invoke a GetValue() request for the **cmi.comments_from_lms._children** data model element.
REQ_94.2	The SCO shall only invoke a GetValue() request for the **cmi.comments_from_lms._count** data model element.
REQ_94.3	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.comments_from_lms.n.comment** data model element.
REQ_94.4	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.comments_from_lms.n.location** data model element.
REQ_94.5	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.comments_from_lms.n.timestamp** data model element.
REQ_94.6	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of comments being maintained by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_95	If utilizing the **cmi.completion_status** the SCO shall adhere to the following requirements.
REQ_95.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.completion_status** data model element.
REQ_95.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `unknown` • `completed` • `incomplete` • `not attempted` **ADL NOTE:** The default value that is assumed by the LMS is `unknown`.
REQ_96	If utilizing the **cmi.completion_threshold** the SCO shall adhere to the following requirements.
REQ_96.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.completion_threshold** data model element.

REQ ID	Requirement
REQ_97	If utilizing the **cmi.credit** the SCO shall adhere to the following requirements.
REQ_97.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.credit** data model element.
REQ_98	If utilizing the **cmi.entry** the SCO shall adhere to the following requirements.
REQ_98.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.entry** data model element.
REQ_99	If utilizing the **cmi.exit** the SCO shall adhere to the following requirements.
REQ_99.1	The SCO, if needed, shall only invoke a SetValue() request for the **cmi.exit** data model element.
REQ_99.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `""` (empty characterstring) • `normal` • `suspend` • `time-out` • `logout` **ADL Note:** This value is being deprecated and should not be used. **ADL NOTE:** The default value that is assumed by the LMS is "" (empty characterstring).
REQ_100	If utilizing the **cmi.interactions** the SCO shall adhere to the following requirements.
REQ_100.1	The SCO shall only invoke a GetValue() request for the **cmi.interactions._children** data model element.
REQ_100.2	The SCO shall only invoke a GetValue() request for the **cmi.interactions._count** data model element.
REQ_100.3	During a SetValue() request, the SCO shall ensure that the index value (n) is either less than or equal to the current number of interaction records being maintained by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. Arrays are treated as packed arrays and have to be set in order.
REQ_100.4	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of interaction records being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_100.5	If utilizing the **cmi.interactions.n.id** the SCO shall adhere to the following requirements.
REQ_100.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.id** data model element.
REQ_100.5.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.id** value is a valid long_identifier_type.
REQ_100.5.3	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.id** value is unique at least within the scope of the SCO.

REQ ID	Requirement
REQ_100.5.4	The SCO shall ensure that the **cmi.interactions.n.id** is set first for each interaction record that the SCO wishes to store.
REQ_100.6	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.type** data model element.
REQ_100.6.1	If the SCO is going to store **cmi.interactions.n.correct_responses** and/or **cmi.interactions.n.learner_response**, the SCO shall ensure that the **cmi.interactions.n.type** is set prior to the **cmi.interactions.n.correct_responses** and/or **cmi.interactions.n.learner_response**.
REQ_100.6.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.type** is one of the following restricted tokens: • `true-false` • `choice` • `fill-in` • `long-fill-in` • `likert` • `matching` • `performance` • `sequencing` • `numeric` • `other`
REQ_100.7	If utilizing the **cmi.interactions.n.objectives** the SCO shall adhere to the following requirements.
REQ_100.7.1	The SCO shall only invoke a GetValue() request for the **cmi.interactions.n.objectives._count** data model element.
REQ_100.7.2	If utilizing the **cmi.interactions.n.objectives.m.id** the SCO shall adhere to the following requirements.
REQ_100.7.2.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.objectives.m.id** data model element.
REQ_100.7.2.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.objectives.m.id** value is a valid long_identifier_type.
REQ_100.7.2.3	During a SetValue() request, the SCO shall ensure that the index value (n) is either less than or equal to the current number of objective identifiers being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. Arrays are treated as packed arrays and have to be set in order.
REQ_100.7.2.4	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of objective identifiers being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_100.8	If utilizing the **cmi.interactions.n.timestamp** the SCO shall adhere to the following requirements.
REQ_100.8.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.timestamp** data model element.

REQ ID	Requirement
REQ_100.8.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.timestamp** value is a valid time (second,10,0).
REQ_100.9	If utilizing the **cmi.interactions.n.correct_responses** the SCO shall adhere to the following requirements.
REQ_100.9.1	The SCO shall only invoke a GetValue() request for the **cmi.interactions.n.correct_responses._count** data model element.
REQ_100.9.2	If utilizing the **cmi.interactions.n.correct_responses.m.pattern** the SCO shall adhere to the following requirements.
REQ_100.9.2.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.correct_responses.m.pattern** data model element.
REQ_100.9.2.2	If the SCO is going to store **cmi.interactions.n.correct_responses**, the SCO shall ensure that the **cmi.interactions.n.type** is set prior to the **cmi.interactions.n.correct_responses.m.pattern**.
REQ_100.9.2.3	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.correct_responses.m.pattern** value is a valid correct response type that is based on the **cmi.interactions.n.type**. **ADL NOTE:** Refer to the Refer to *Section 2.1.4 Run-Time Environment Data Model Data Type Conformance Requirements* for more information.
REQ_100.9.2.3.1	If setting the **cmi.interactions.n.correct_response.m.pattern** for a `true-false` interaction type, the SCO shall only set one correct response pattern (i.e., only one entry in the pattern array is permitted – **cmi.interactions.n.correct_response.*m*.pattern**, ***m*** can only equal 0).
REQ_100.9.2.3.2	If setting the **cmi.interactions.n.correct_response.m.pattern** for a `likert` interaction type, the SCO shall only set one correct response pattern (i.e., only one entry in the pattern array is permitted – **cmi.interactions.n.correct_response.*m*.pattern**, ***m*** can only equal 0).
REQ_100.9.2.3.3	If setting the **cmi.interactions.n.correct_response.m.pattern** for a `numeric` interaction type, the SCO shall only set one correct response pattern (i.e., only one entry in the pattern array is permitted – **cmi.interactions.n.correct_response.*m*.pattern**, ***m*** can only equal 0).
REQ_100.9.2.4	During a SetValue() request, the SCO shall ensure that the index value (n) is either less than or equal to the current number of correct responses being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. Arrays are treated as packed arrays and have to be set in order.
REQ_100.9.2.5	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of correct responses being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_100.10	If utilizing the **cmi.interactions.n.weighting** the SCO shall adhere to the following requirements.
REQ_100.10.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.weighting** data model element.
REQ_100.10.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.weighting** value is a valid real(10,7).

REQ ID	Requirement
REQ_100.11	If utilizing the **cmi.interactions.n.learner_response** the SCO shall adhere to the following requirements.
REQ_100.11.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.learner_response** data model element.
REQ_100.11.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.learner_response** value is a valid learner response type that is based on the **cmi.interactions.n.type**. **ADL NOTE:** Refer to *Section 2.1.4 Run-Time Environment Data Model Data Type Conformance Requirements* for more information.
REQ_100.11.3	If the SCO is going to store **cmi.interactions.n.learner_response**, the SCO shall ensure that the **cmi.interactions.n.type** is set prior to the **cmi.interactions.n.learner_response**.
REQ_100.12	If utilizing the **cmi.interactions.n.result** the SCO shall adhere to the following requirements.
REQ_100.12.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.result** data model element.
REQ_100.12.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `correct` • `incorrect` • `unanticipated` • `neutral` • real(10,7)
REQ_100.13	If utilizing the **cmi.interactions.n.latency** the SCO shall adhere to the following requirements.
REQ_100.13.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.latency** data model element.
REQ_100.13.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.latency** value is a valid timeinterval (second,10,2).
REQ_100.14	If utilizing the **cmi.interactions.n.description** the SCO shall adhere to the following requirements.
REQ_100.14.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.interactions.n.description** data model element.
REQ_100.14.2	During a SetValue() request, the SCO shall ensure that the **cmi.interactions.n.description** value is a valid localized_string_type.
REQ_101	If utilizing the **cmi.launch_data** the SCO shall adhere to the following requirements.
REQ_101.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.launch_data** data model element.
REQ_102	If utilizing the **cmi.learner_id** the SCO shall adhere to the following requirements.

REQ ID	Requirement
REQ_102.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.learner_id** data model element.
REQ_103	If utilizing the **cmi.learner_name** the SCO shall adhere to the following requirements.
REQ_103.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.learner_name** data model element.
REQ_104	If utilizing the **cmi.learner_preference** the SCO shall adhere to the following requirements.
REQ_104.1	The SCO shall only invoke a GetValue() request for the **cmi.learner_preference._children** data model element.
REQ_104.2	If utilizing the **cmi.learner_preference.audio_level** the SCO shall adhere to the following requirements.
REQ_104.2.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.learner_preference.audio_level** data model element.
REQ_104.2.2	During a SetValue() request, the SCO shall ensure that the **cmi.learner_preference.audio_level** value is a valid real(10,7).
REQ_104.2.3	During a SetValue() request, the SCO shall ensure that the **cmi.learner_preference.audio_level** value is real number greater than or equal to 0.
REQ_104.3	If utilizing the **cmi.learner_preference.language** the SCO shall adhere to the following requirements.
REQ_104.3.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.learner_preference.language** data model element.
REQ_104.3.2	During a SetValue() request, the SCO shall ensure that the **cmi.learner_preference.language** value is a valid language_type or an empty characterstring ("").
REQ_104.4	If utilizing the **cmi.learner_preference.delivery_speed** the SCO shall adhere to the following requirements.
REQ_104.4.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.learner_preference.delivery_speed** data model element.
REQ_104.4.2	During a SetValue() request, the SCO shall ensure that the **cmi.learner_preference.delivery_speed** value is a valid real(10,7).
REQ_104.4.3	During a SetValue() request, the SCO shall ensure that the **cmi.learner_preference.delivery_speed** value is real number greater than or equal to 0.
REQ_104.5	If utilizing the **cmi.learner_preference.audio_captioning** the SCO shall adhere to the following requirements.
REQ_104.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.learner_preference.audio_captioning** data model element.

REQ ID	Requirement
REQ_104.5.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `-1` • `0` • `1` **ADL NOTE:** The default value that is assumed by the LMS is `0`.
REQ_105	If utilizing the **cmi.location** the SCO shall adhere to the following requirements.
REQ_105.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.location** data model element.
REQ_105.2	During a SetValue() request, the SCO shall ensure that the **cmi.location** value is a valid characterstring.
REQ_106	If utilizing the **cmi.max_time_allowed** the SCO shall adhere to the following requirements.
REQ_106.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.max_time_allowed** data model element.
REQ_107	If utilizing the **cmi.mode** the SCO shall adhere to the following requirements.
REQ_107.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.mode** data model element.
REQ_108	If utilizing the **cmi.objectives** the SCO shall adhere to the following requirements.
REQ_108.1	The SCO shall only invoke a GetValue() request for the **cmi.objectives._children** data model element.
REQ_108.2	The SCO shall only invoke a GetValue() request for the **cmi.objectives._count** data model element.
REQ_108.3	During a SetValue() request, the SCO shall ensure that the index value (n) is either less than or equal to the current number of objective records being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. Arrays are treated as packed arrays and have to be set in order.
REQ_108.4	During a GetValue() request, the SCO shall ensure that the index value (n) is less than the current number of objective records being stored by the LMS. **ADL NOTE:** Arrays are implemented as zero-based. SCOs are not permitted to retrieve an index that was never set.
REQ_108.5	If utilizing the **cmi.objectives.n.id** the SCO shall adhere to the following requirements.
REQ_108.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.id** data model element.
REQ_108.5.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.id** value is a valid long_identifier_type.
REQ_108.5.3	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.id** value is unique at least within the scope of the SCO.

REQ ID	Requirement
REQ_108.5.4	The SCO shall ensure that the **cmi.objectives.n.id** is set first for each objective record that the SCO wishes to store.
REQ_108.5.5	The SCO shall ensure that the **cmi.objectives.n.id** is only set once and if set more than once the value for parameter_2 cannot be a different value than what is currently being stored by the LMS.
REQ_108.6	If utilizing the **cmi.objectives.n.score** the SCO shall adhere to the following requirements.
REQ_108.6.1	The SCO shall only invoke a GetValue() request for the **cmi.objectives.n.score._children** data model element.
REQ_108.6.2	If utilizing the **cmi.objectives.n.score.scaled** the SCO shall adhere to the following requirements.
REQ_108.6.2.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.score.scaled** data model element.
REQ_108.6.2.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.score.scaled** value is a valid real(10,7).
REQ_108.6.2.3	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.score.scaled** value is in the range of -1.0 to 1.0.
REQ_108.6.3	If utilizing the **cmi.objectives.n.score.raw** the SCO shall adhere to the following requirements.
REQ_108.6.3.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.score.raw** data model element.
REQ_108.6.3.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.score.raw** value is a valid real(10,7).
REQ_108.6.4	If utilizing the **cmi.objectives.n.score.min** the SCO shall adhere to the following requirements.
REQ_108.6.4.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.score.min** data model element.
REQ_108.6.4.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.score.min** value is a valid real(10,7).
REQ_108.6.5	If utilizing the **cmi.objectives.n.score.max** the SCO shall adhere to the following requirements.
REQ_108.6.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.score.max** data model element.
REQ_108.6.5.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.score.max** value is a valid real(10,7).
REQ_108.7	If utilizing the **cmi.objectives.n.success_status** the SCO shall adhere to the following requirements.
REQ_108.7.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.success_status** data model element.

REQ ID	Requirement
REQ_108.7.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `unknown` • `passed` • `failed` **ADL NOTE:** The default value that is assumed by the LMS is `unknown`.
REQ_108.8	If utilizing the **cmi.objectives.n.completion_status** the SCO shall adhere to the following requirements.
REQ_108.8.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.completion_status** data model element.
REQ_108.8.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `unknown` • `completed` • `incomplete` • `not attempted` **ADL NOTE:** The default value that is assumed by the LMS is `unknown`.
REQ_108.9	If utilizing the **cmi.objectives.n.description** the SCO shall adhere to the following requirements.
REQ_108.9.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.description** data model element.
REQ_108.9.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.description** value is a valid localized_string_type.
REQ_108.10	If utilizing the **cmi.objectives.n.progress_measure** the SCO shall adhere to the following requirements.
REQ_108.10.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.objectives.n.progress_measure** data model element.
REQ_108.10.2	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.progress_measure** value is a valid real(10,7).
REQ_108.10.3	During a SetValue() request, the SCO shall ensure that the **cmi.objectives.n.progress_measure** value is in the range of 0.0 to 1.0 (inclusive).
REQ_109	If utilizing the **cmi.progress_measure** the SCO shall adhere to the following requirements.
REQ_109.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.progress_measure** data model element.
REQ_109.2	During a SetValue() request, the SCO shall ensure that the **cmi.progress_measure** value is a valid real(10,7).
REQ_109.3	During a SetValue() request, the SCO shall ensure that the **cmi.progress_measure** value is in the range of 0.0 to 1.0 (inclusive).
REQ_110	If utilizing the **cmi.scaled_passing_score** the SCO shall adhere to the following requirements.

REQ ID	Requirement
REQ_110.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.scaled_passing_score** data model element.
REQ_111	If utilizing the cmi.score the SCO shall adhere to the following requirements.
REQ_111.1	The SCO shall only invoke a GetValue() request for the **cmi.score._children** data model element.
REQ_111.2	If utilizing the **cmi.score.scaled** the SCO shall adhere to the following requirements.
REQ_111.2.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.score.scaled** data model element.
REQ_111.2.2	During a SetValue() request, the SCO shall ensure that the **cmi.score.scaled** value is a valid real(10,7).
REQ_111.2.3	During a SetValue() request, the SCO shall ensure that the **cmi.score.scaled** value is in the range of -1.0 to 1.0.
REQ_111.3	If utilizing the **cmi.score.raw** the SCO shall adhere to the following requirements.
REQ_111.3.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.score.raw** data model element.
REQ_111.3.2	During a SetValue() request, the SCO shall ensure that the **cmi.score.raw** value is a valid real(10,7).
REQ_111.4	If utilizing the **cmi.score.min** the SCO shall adhere to the following requirements.
REQ_111.4.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.score.min** data model element.
REQ_111.4.2	During a SetValue() request, the SCO shall ensure that the **cmi.score.min** value is a valid real(10,7).
REQ_111.5	If utilizing the **cmi.score.max** the SCO shall adhere to the following requirements.
REQ_111.5.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.score.max** data model element.
REQ_111.5.2	During a SetValue() request, the SCO shall ensure that the **cmi.score.max** value is a valid real(10,7).
REQ_112	If utilizing the **cmi.session_time** the SCO shall adhere to the following requirements.
REQ_112.1	The SCO, if needed, shall only invoke a SetValue() request for the **cmi.session_time** data model element.
REQ_112.2	During a SetValue() request, the SCO shall ensure that the **cmi.session_time** value is a valid timeinterval (second,10,2).
REQ_113	If utilizing the **cmi.success_status** the SCO shall adhere to the following requirements.
REQ_113.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **cmi.success_status** data model element.
REQ_113.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `unknown` • `passed` • `failed` **ADL NOTE:** The default value that is assumed by the LMS is `unknown`.
REQ_114	If utilizing the **cmi.suspend_data** the SCO shall adhere to the following requirements.
REQ_114.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the

REQ ID	Requirement
	cmi.suspend_data data model element.
REQ_114.2	During a SetValue() request, the SCO shall ensure that the **cmi.suspend_data** value is a valid characterstring.
REQ_115	If utilizing the **cmi.time_limit_action** the SCO shall adhere to the following requirements.
REQ_115.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.time_limit_action** data model element.
REQ_116	If utilizing the **cmi.total_time** the SCO shall adhere to the following requirements.
REQ_116.1	The SCO, if needed, shall only invoke a GetValue() request for the **cmi.total_time** data model element.

4.1.4. Run-Time Environment Data Model Data Type Conformance Requirements

Refer to *Section 2.1.4: Run-Time Environment Data Model Data Type Conformance Requirements* for details on the requirements that SCOs must adhere to when using the various data model elements as it applies the data model element types.

4.1.5. Run-Time Navigation Data Model Conformance Requirements

LMSs are required to implement all SCORM Run-Time Navigation Data Model elements to be considered SCORM 2004 3rd Edition conformant.

There is no requirement that a SCO use the Run-Time Navigation Data Model. If the SCO does implement the ability to exchange navigation data model elements with an LMS, it must use the defined SCORM Run-Time Navigation Data Model elements, within the bounds of the requirements as defined in Table 4.1.5a.

The SCO shall adhere to the requirements defined in Table 4.1.5a to be considered conformant to the SCO RTE 1.0 conformance category.

Table 4.1.5a: SCO Run-Time Navigation Data Model Conformance Requirements

REQ ID	Requirement
REQ_51	The SCO shall adhere to the following requirements.
REQ_51.1	The SCO, if needed, shall invoke a GetValue() or SetValue() request for the **adl.nav.request** data model element.
REQ_51.2	During a SetValue() request, the SCO shall ensure that the value to be used for setting is one of the following reserved tokens: • `continue` • `previous`

REQ ID	Requirement
	• `choice` • `exit` • `exitAll` • `abandon` • `abandonAll` • `_none_`
REQ_51.2.1	During a SetValue() request for a choice navigation request, the SCO shall ensure that the value to be used for setting of the following format: • {target=<*STRING*>}choice The value of the <*STRING*> should reference the **identifier** attribute of an **<imscp:item>** element from the content package manifest, which was used to derive the "Activity Tree".
REQ_52	The SCO shall adhere to the following requirements.
REQ_52.1	The SCO, if needed, shall only invoke a GetValue() request for the **adl.nav.request_valid.continue** data model element.
REQ_53	The SCO shall adhere to the following requirements.
REQ_53.1	The SCO, if needed, shall only invoke a GetValue() request for the **adl.nav.request_valid.previous** data model element.
REQ_54	The SCO shall adhere to the following requirements.
REQ_54.1	The SCO, if needed, shall only invoke a GetValue() request for the **adl.nav.request_valid.choice.{target=<*STRING*>}** data model element.

APPENDIX A
Sequencing Conformance Requirements

This page intentionally left blank.

Sequencing Conformance Requirements

SCORM does not state explicit requirements for the look and feel of the user interface in cases where the LMS cannot identify some content to launch. In the context of sequencing, this means that the sequencing conformance tests do not include any "negative" (failure) test cases whwew the LMS's sequencing implementation identifies a pseudo-code exception. It is recommend that if a failure occurs, the LMS should attempt to handle that failure in a way that minimizes its impact on the learner while retaining as much sequencing state as possible.

Paths through the sequencing pseudo-code that explicitly end in an exception being identified or that end in no activity being identified for delivery are not tested explicitly. However, some of the User Interface interoperability tests are intended to minimize the occurrence of such exceptions by preventing the learner from triggering Navigation Events that do not identify an activity for delivery.

The launch behavior defined in the SCORM RTE prohibits "real" sequencing from occurring prior to the current content object being taken away. Therefore, paths through the pseudo-code that test if the current activity is "active" are not tested.

Table A1 lists the sequencing aspects not directly tested.

Table A1: Sequencing Aspects Not Tested

Disabled Pre Condition Sequencing Rule in the context of a Continue or Previous Navigation Event	*Limit Condition* violations in the context of a Continue or Previous Navigation Event	A *Limit Condition* of *Attempt Limit* equal to Zero
Sequencing Control Choice Exit on a cluster that also has Sequencing Control Choice Flow equal to True	Selection Controls	Randomization Controls
Exit All Post Condition Sequencing Rule		

In addition, SCORM does not define when or how an LMS should perform validation of navigation requests. Therefore the navigation data model (`adl.nav.xxx_valid`) elements are only tested to ensure they are implemented and return the defined vocabulary. The values returned from those requests are not tested against a known "correct" result.

Test Case Explanation

Test Case

This label uniquely identifies the test case.

Activity Tree Structure

This is a graphical representation of the Activity Tree used for the test case, where circles represent the individual activities in the activity tree. Darker circles indicate cluster activities.

Sequencing Information

This table describes the sequencing information (Sequencing Definition Model elements) applied to each activity in the test case's Activity Tree. If the value of a specific Sequencing Definition Model element is not explicitly defined in the table, its default value as defined in the SCORM Sequencing and Navigation applies.

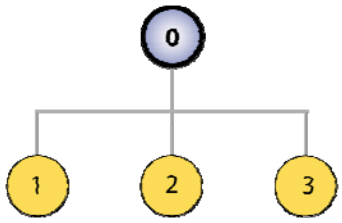
Test Script

Each test is executed in the context of an anonymous learner because the sequencing behaviors are unconcerned with the exact identity of the learner. All tests assume that no tracking information has been recorded for any activity in the test case's Activity Tree; in other words, the root activity in the Activity tree has not been attempted yet. Also, all tests assume that no global shared objects currently exist in the system for the anonymous learner.

The test script describes a series of actions applied to the test case's Activity Tree and the expected results of those actions. If at any step of the test script, the LMS does not exhibit the expected results, the LMS is considered non-conformant and the test stops.

Test Case: CM-01

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Limit Conditions: Attempt Absolute Duration Limit == P5Y6M4DT12H30M58S
2	Objectives: Primary Objective: Satisfied By Measure == true Minimum Normalized Measure == 0.8
3	Limit Conditions: Attempt Absolute Duration Limit == P5Y6M4DT12H30M58S Objectives: Primary Objective: Satisfied By Measure == true Minimum Normalized Measure == 0.7

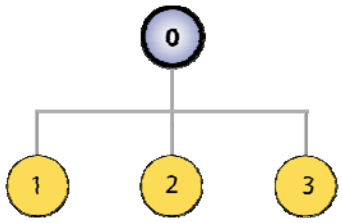
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Process a *Previous* navigation request	Identify Activity 2 for delivery
5.	Process a *Continue* navigation request	Identify Activity 3 for delivery
6.	Process a *Previous* navigation request	Identify Activity 2 for delivery
7.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: CM-02a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Objectives: Primary Objective: *empty* Sequencing Rules: Pre Condition Rule: If satisfied, then skip
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ

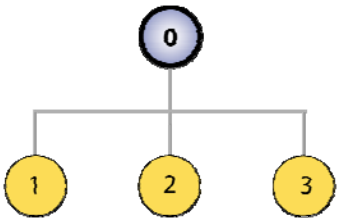
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Process a *Previous* navigation request	Identify Activity 1 for delivery
5.	Process a *Continue* navigation request	Identify Activity 3 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: CM-02b

Activity Tree Structure:



Sequencing Information:

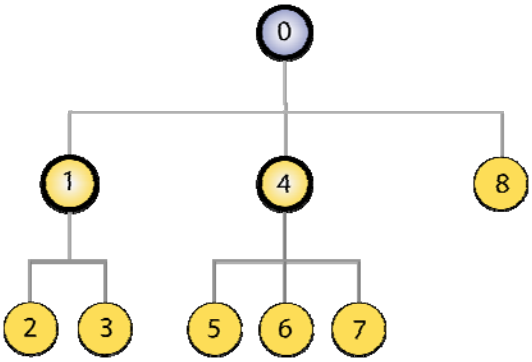
Activity	Sequencing Information
0	Control Mode: Flow == true Control Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Objective: Objective ID == obj2 Objective: Objective ID == obj3
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj1 Delivery Controls: Objective Set by Content == true
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Process a *Previous* navigation request	Identify Activity 2 for delivery
5.	Set Activity 2's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 3 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery
7.	Process a *Continue* navigation request	Identify Activity 3 for delivery
8.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: CM-03a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true Choice == false
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj2 Objective: Objective ID == obj3 Objective: Objective ID == obj4 Objective: Objective ID == obj5
4	Control Mode: Flow == true Choice == false Forward Only == true
5	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip
6	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip

Activity	Sequencing Information
7	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip
8	Default

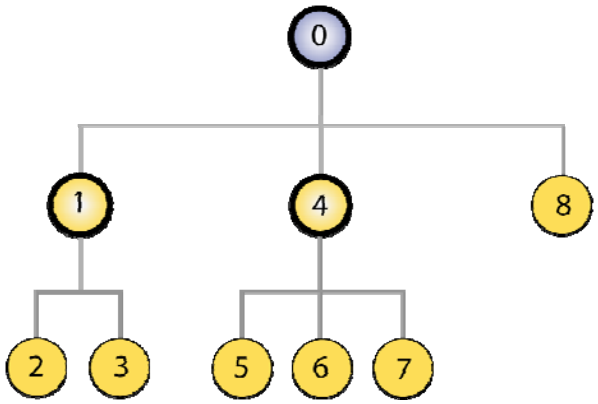
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Set Activity 6's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 7 for delivery
6.	Set Activity 7's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 8 for delivery
7.	Process a *Previous* navigation request	Identify Activity 5 for delivery

Test Case: CM-03b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Model: Flow == true Choice == false
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj1
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID ==obj2 Objective: Objective ID == obj3 Objective: Objective ID == obj4 Objective: Objective ID == obj5 Objective: Objective ID == obj6
4	Control Mode: Flow == true Choice == false Forward Only ==true

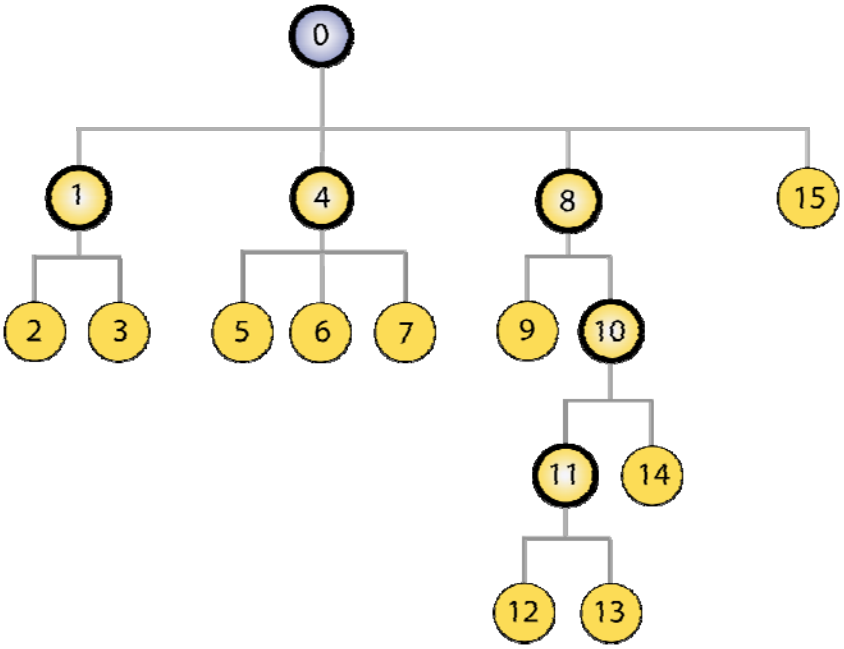
Activity	Sequencing Information
5	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip
6	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip
7	Sequencing Rules: Pre Condition Rule: If attempted and not completed, then skip
8	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.completion_status to `incomplete` Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Set Activity 6's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 7 for delivery
6.	Set Activity 7's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 8 for delivery
7.	Process a *Previous* navigation request	Identify Activity 3 for delivery
8	Process a *Previous* navigation request	Identify Activity 2 for delivery

Test Case: CM-04a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Default
1	Default
2	Default
3	Default
4	Default
5	Default
6	Default
7	Default
8	Default
9	Default
10	Default
11	Control Mode: Flow == true
12	Default
13	Default

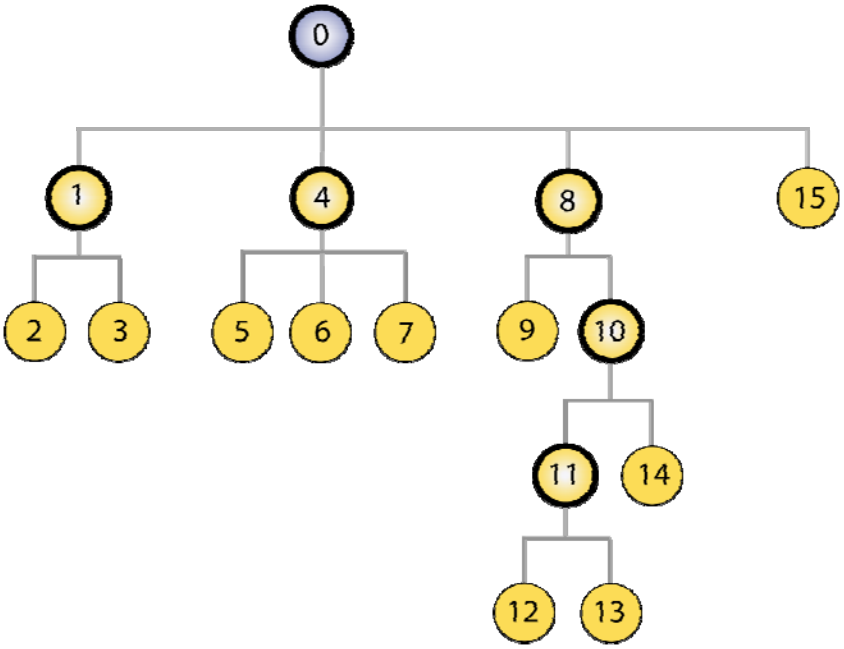
14	Default
15	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Choice* navigation request for Activity 12	Identify Activity 12 for delivery
2.	Process a *Choice* navigation request for Activity 12	Identify Activity 12 for delivery
3.	Process a *Choice* navigation request for Activity 11	Identify Activity 12 for delivery
4.	Process a *Choice* navigation request for Activity 5	Identify Activity 5 for delivery
5.	Process a *Choice* navigation request for Activity 7	Identify Activity 7 for delivery
6.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
7.	Process a *Choice* navigation request for Activity 15	Identify Activity 15 for delivery
8.	Process a *Choice* navigation request for Activity 9	Identify Activity 9 for delivery
9.	Process a *Choice* navigation request for Activity 14	Identify Activity 14 for delivery
10	Process a *Choice* navigation request for Activity 3	Identify Activity 3 for delivery

Test Case: CM-04b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true
1	Control Mode: Flow == true
2	Default
3	Default
4	Default
5	Default
6	Default
7	Default
8	Default
9	Default
10	Control Mode: Flow == true
11	Control Mode:

	Flow == true Sequencing Rules: Pre Condition Rule: If attempted, then skip
12	Default
13	Default
14	Default
15	Default

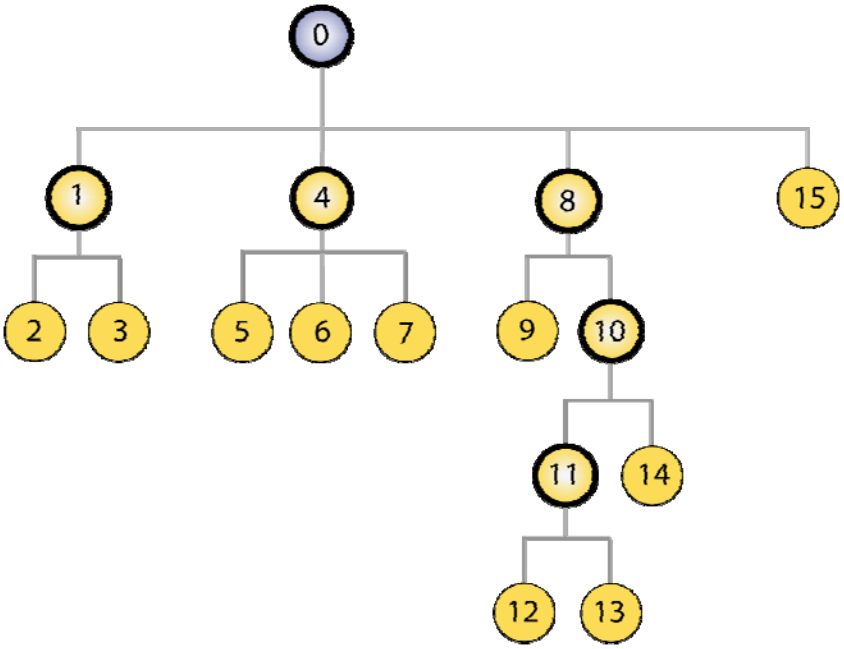
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Choice* navigation request for Activity 11	Identify Activity 12 for delivery
3.	Process a *Choice* navigation request for Activity 10	Identify Activity 14 for delivery
4.	Process a *Choice* navigation request for Activity 0	Identify Activity 2 for delivery
5.	Process a *Choice* navigation request for Activity 15	Identify Activity 15 for delivery
6.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
7.	Process an *Exit All* navigation Request	End the Sequencing Session

Test Case: CM-04c

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Default
1	Control Mode: Flow == true
2	Default
3	Default
4	Default
5	Default
6	Default
7	Default
8	Default
9	Default
10	Control Mode: Flow == true
11	Control Mode: Flow == true

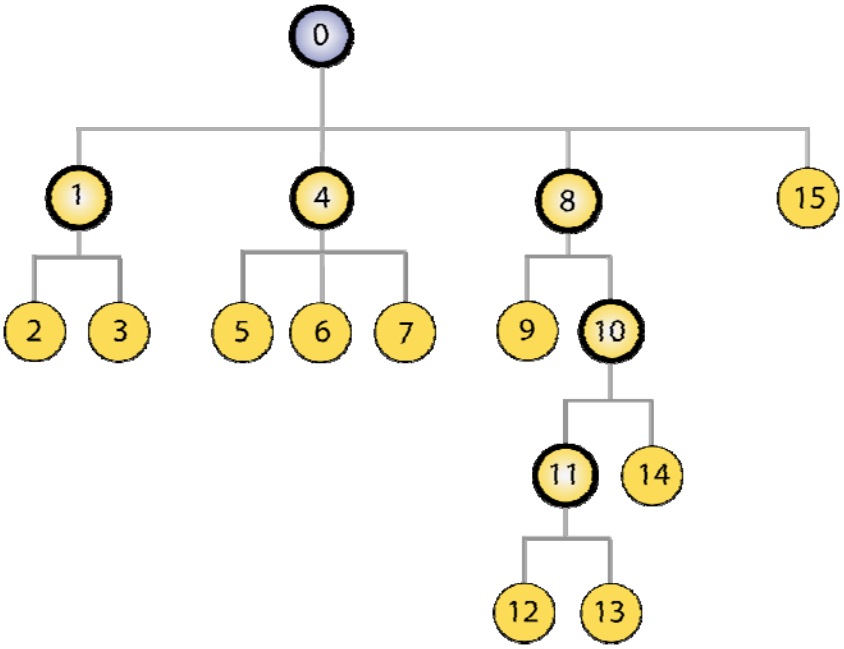
	Sequencing Rules: Pre Condition Rule: If attempted, then skip
12	Default
13	Default
14	Default
15	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Choice* navigation request for Activity 10	Identify Activity 12 for delivery
2.	Process a *Choice* navigation request for Activity 11	Identify Activity 12 for delivery
3.	Process a *Choice* navigation Request for Activity 1	Identify Activity 2 for delivery
4.	Process a *Choice* navigation Request for Activity 10	Identify Activity 14 for delivery
5.	Process a *Choice* navigation Request for Activity 13	Identify Activity 13 for delivery
6.	Process an *Exit All* navigation Request	End the Sequencing Session

Test Case: CM-04d

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Default
1	Default
2	Default
3	Default
4	Default
5	Default
6	Default
7	Default
8	Default
9	Default
10	Default
11	Default
12	Default
13	Default
14	Default

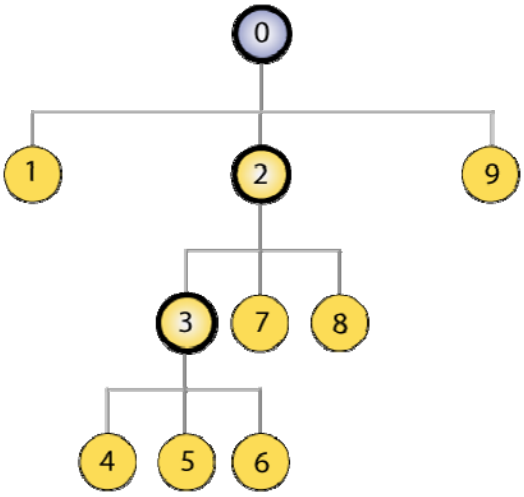
15	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Choice* navigation request for Activity 2	Identify Activity 2 for delivery
2.	Process a *Choice* navigation request for Activity 15	Identify Activity 15 for delivery
3.	Process a *Choice* navigation request for Activity 13	Identify Activity 13 for delivery
4.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
5.	Process a *Choice* navigation request for Activity 9	Identify Activity 9 for delivery
6.	Process a *Choice* navigation request for Activity 3	Identify Activity 3 for delivery
7.	Process a *Choice* navigation request for Activity 12	Identify Activity 12 for delivery
8.	Process a *Choice* navigation request for Activity 5	Identify Activity 5 for delivery
9.	Process a *Choice* navigation request for Activity 7	Identify Activity 7 for delivery
10	Process a *Choice* navigation request for Activity 14	Identify Activity 14 for delivery

Test Case: CM-05

Activity Tree Structure:



Sequencing Information:

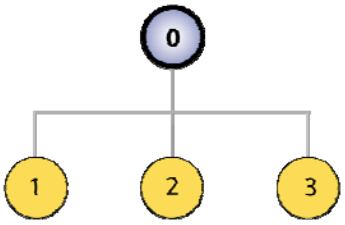
Activity	Sequencing Information
0	Control Mode: Flow == true
1	Default
2	Control Mode: Flow == true
3	Control Mode: Flow == true
4	Default
5	Default
6	Default
7	Default
8	Default
9	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
3.	Process a *Suspend All* navigation request	End Sequencing Session
4.	Process a *Resume All* navigation request	Identify Activity 6 for delivery
5.	Process a *Choice* navigation request for Activity 8	Identify Activity 8 for delivery
6.	Process a *Suspend All* navigation request	End Sequencing Session
7.	Process a *Resume All* navigation request	Identify Activity 8 for delivery

Test Case: CM-06

Activity Tree Structure:



Sequencing Information:

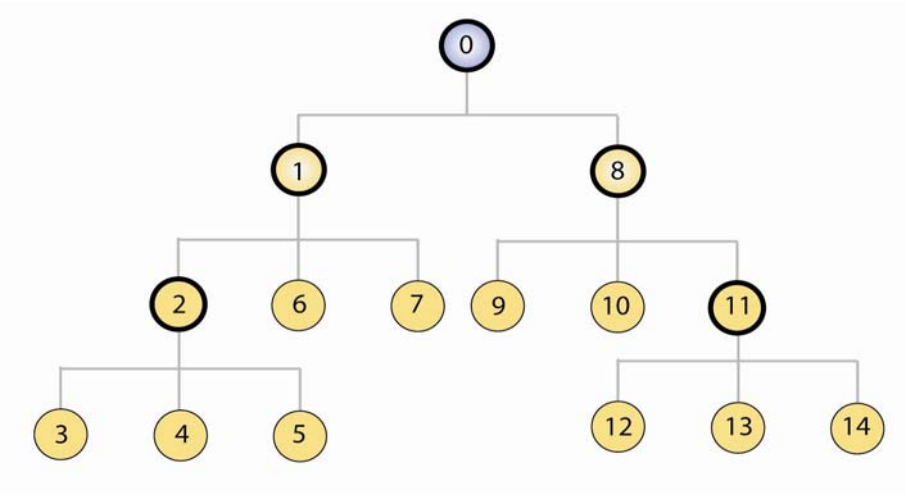
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Default
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Process a *Continue* navigation request	End sequencing session.

Test Case: CM-07a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true
1	Control Mode: Flow == true Choice Exit == false
2	Control Mode: Flow == true
3	Default
4	Default
5	Sequencing Rules: Pre Condition Rule: If always, then disabled
6	Control Mode: Choice Exit == false
7	Default
8	Control Mode: Flow == true Choice Exit == false
9	Default
10	Control Mode: Choice Exit == false
11	Control Mode: Flow == true
12	Default
13	Default
14	Sequencing Rules:

	Pre Condition Rule: If always, then disabled

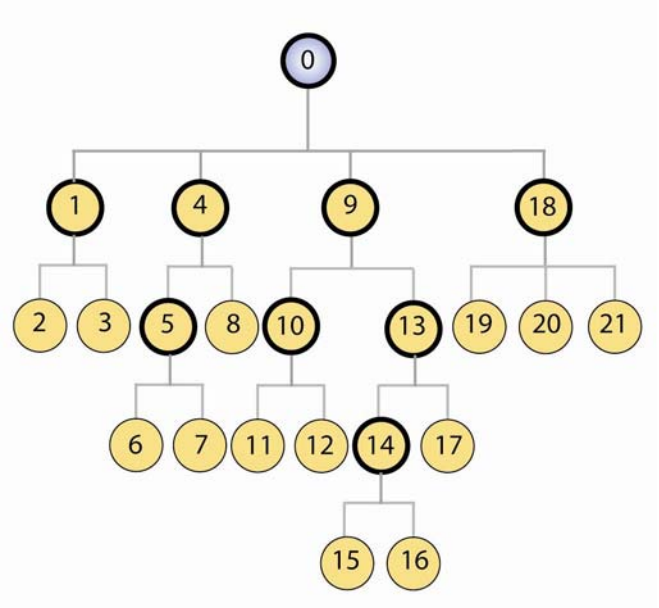
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 3 for delivery
2.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
3.	Process a *Continue* navigation request	Identify Activity 7 for delivery
4.	Process a *Continue* navigation request	Identify Activity 9 for delivery
5.	Process a *Choice* navigation request for Activity 13	Identify Activity 13 for delivery
6.	Process a *Choice* navigation request for Activity 10	Identify Activity 10 for delivery
7.	Process a *Continue* navigation request	Identify Activity 12 for delivery
8.	Process a *Continue* navigation request	Identify Activity 13 for delivery

Test Case: CM-07b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Default
1	Control Mode: Flow == true
2	Sequencing Rules: Pre Condition Rule: If always, then skip
3	Default
4	Control Mode: Flow == true Sequencing Rules: Pre Condition Rule: If always, then disabled
5	Control Mode: Flow == true
6	Default
7	Default
8	Default
9	Control Mode: Flow == true
10	Control Mode: Flow == true
11	Default

12	Default
13	Control Mode: Flow == true
14	Control Mode: Flow == true Forward Only == true
15	Sequencing Rules: Pre Condition Rule: If always, then skip
16	Sequencing Rules: Pre Condition Rule: If attempted, then skip
17	Default
18	Default
19	Default
20	Default
21	Default

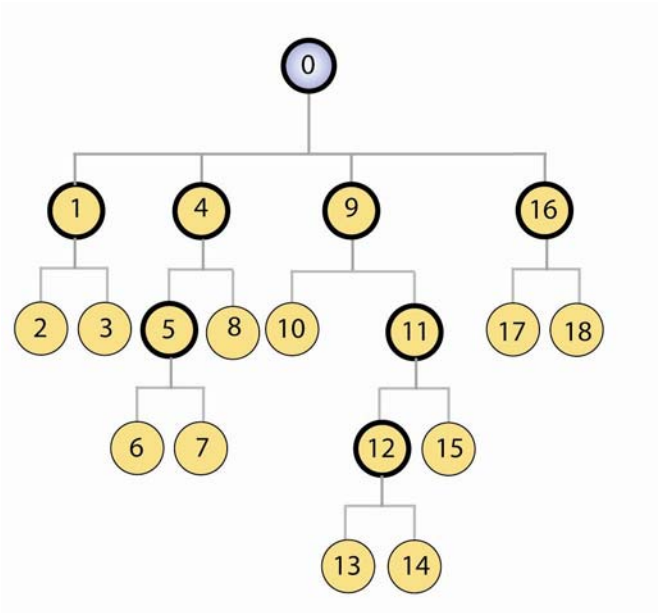
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Choice* navigation request for Activity 3	Identify Activity 3 for delivery
2.	Process a *Choice* navigation request for Activity 16	Identify Activity 16 for delivery
3.	Process a *Choice* navigation request for Activity 17	Identify Activity 17 for delivery
4.	Process a *Choice* navigation request for Activity 11	Identify Activity 11 for delivery
5	Process a *Choice* navigation request for Activity 19	Identify Activity 19 for delivery
6.	Process a *Choice* navigation request for Activity 21	Identify Activity 21 for delivery

Test Case: CM-07c

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true
1	Control Mode: Flow == true
2	Limit Conditions: Attempt limit == 1
3	Default
4	Control Mode: Flow == true Sequencing Rules: Pre Condition Rule: If attempted, then hidden from choice
5	Control Mode: Flow == true
6	Default
7	Sequencing Rules: Pre Condition Rule: If always, then hidden from choice
8	Default
9	Control Mode: Flow == true
10	Default

11	Control Mode: Flow == true Limit Conditions: Attempt limit == 1
12	Control Mode: Flow == true
13	Default
14	Default
15	Default
16	Control Mode: Flow == true
17	Default
18	Sequencing Rules: Pre Condition Rule: If always, then disabled

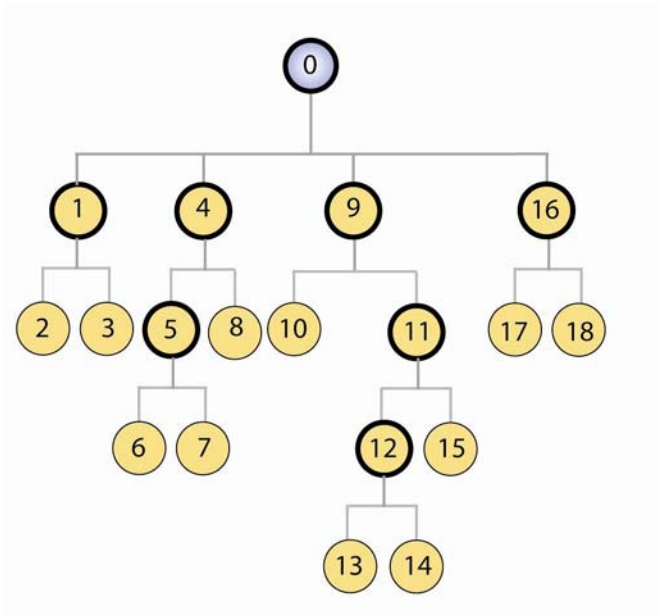
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Choice* navigation request for Activity 8	Identify Activity 8 for delivery
4.	Process a *Choice* navigation request for Activity 13	Identify Activity 13 for delivery
5.	Process a *Choice* navigation request for Activity 17	Identify Activity 17 for delivery

Test Case: CM-07d

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true
2	Default
3	Default
4	Constrain Choice Controls: Constrain Choice == true
5	Default
6	Default
7	Default
8	Default
9	Control Mode: Flow == true Constrain Choice Controls: Constrain Choice == true
10	Default
11	Control Mode: Flow == true Constrain Choice Controls: Constrain Choice == true

12	Control Mode: Flow == true Constrain Choice Controls: Prevent Activation == true
13	Constrain Choice Controls: Constrain Choice == true Prevent Activation == true
14	Default
15	Default
16	Control Mode: Flow == true Constrain Choice Controls: Constrain Choice == true Prevent Activation == true
17	Default
18	Default

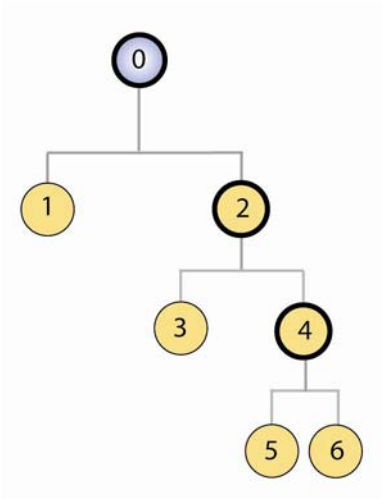
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Choice* navigation request for Activity 3	Identify Activity 3 for delivery
3.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
4.	Process a *Choice* navigation request for Activity 10	Identify Activity 10 for delivery
5.	Process a *Continue* navigation request	Identify Activity 13 for delivery
6.	Process a *Choice* navigation request for Activity 15	Identify Activity 15 for delivery
7.	Process a *Continue* navigation request	Identify Activity 17 for delivery

Test Case: CM-07e

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true
1	Default
2	Control Mode: Flow == true Forward Only == true
3	Default
4	Control Mode: Flow == true Sequencing Rules: Pre Condition Rule: If always, then stop forward traversal
5	Default
6	Default

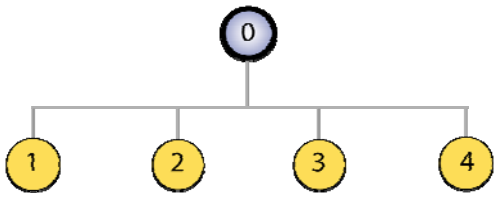
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Choice* navigation request for Activity 3	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
5.	Process a *Choice* navigation request for Activity 0	Identify Activity 1 for delivery

Test Case: CM-07f

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true
1	Sequencing Rule: Pre Condition Rule: If always, then stop forward traversal
2	Default
3	Default
4	Default

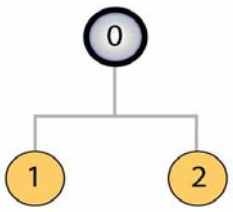
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Choice* navigation request for Activity 4	Identify Activity 4 for delivery

Test Case: CM-08

Activity Tree Structure:



Sequencing Information:

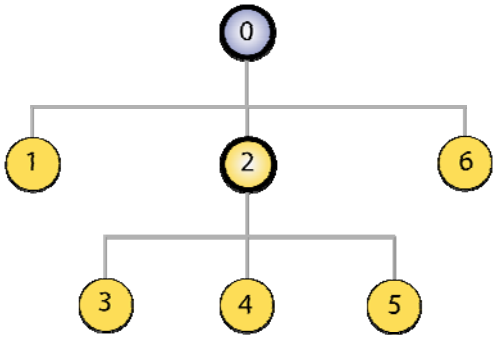
Activity	Sequencing Information
0	Control Mode: Flow == true
1	Sequencing Rule: Post Condition Rule: If always, then Exit All
2	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	End Sequencing Session
3.	Process a *Start* navigation request	Identify Activity 1 for delivery
4.	Process a *Choice* navigation request for Activity 2	End Sequencing Session

Test Case: RU-01aa

Activity Tree Structure:



Sequencing Information:

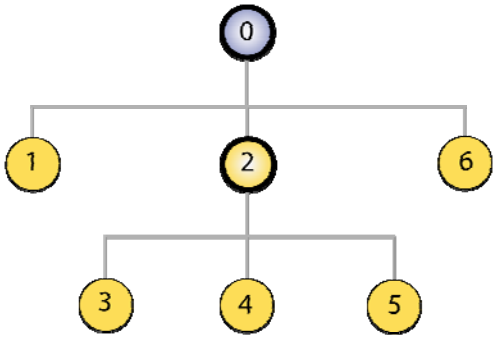
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule If satisfied, then exit Post Condition Rule: If satisfied, then previous
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-01ab

Activity Tree Structure:



Sequencing Information:

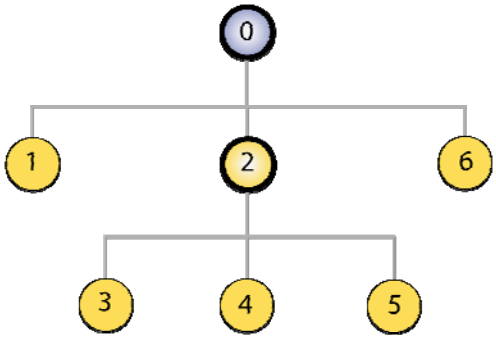
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule If satisfied, then exit Post Condition Rule: If satisfied, then previous
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-01ba

Activity Tree Structure:



Sequencing Information:

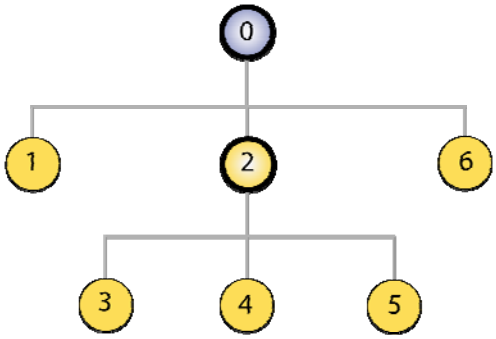
Activity	Sequencing Information
0	Control Mode: Flow == true Control Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-01bb

Activity Tree Structure:



Sequencing Information:

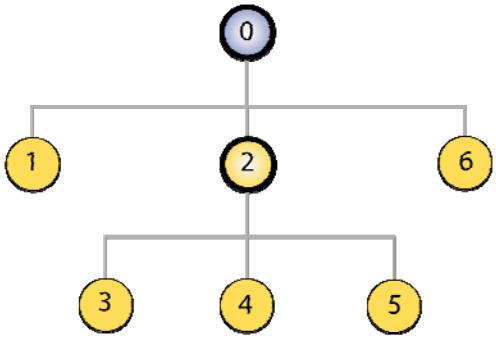
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-02a

Activity Tree Structure:



Sequencing Information:

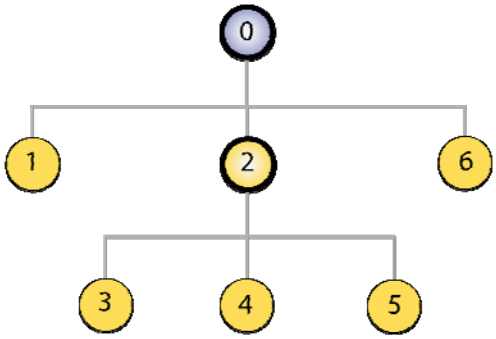
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if any satisfied
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-02b

Activity Tree Structure:



Sequencing Information:

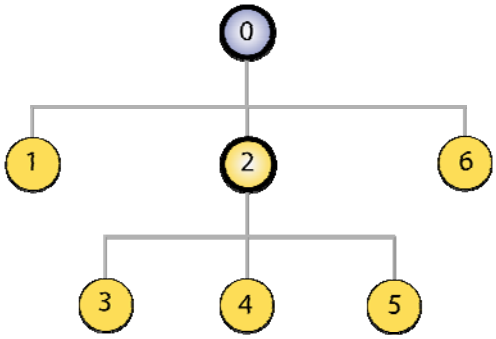
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then continue Rollup Rules: Satisfied if any completed
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Process a *Previous* navigation request	Identify Activity 6 for delivery

Test Case: RU-03a

Activity Tree Structure:



Sequencing Information:

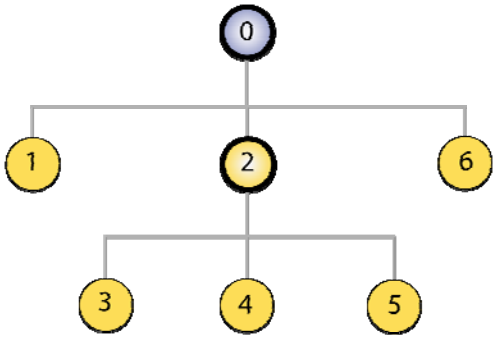
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if at least 1 attempted and not completed
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.completion_status to `incomplete`; Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: RU-03b

Activity Tree Structure



Sequencing Information:

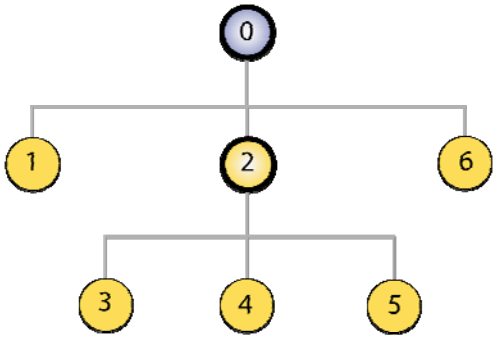
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then previous Rollup Rules: Satisfied if at least 1 attempted and not satisfied
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-04aa

Activity Tree Structure:



Sequencing Information:

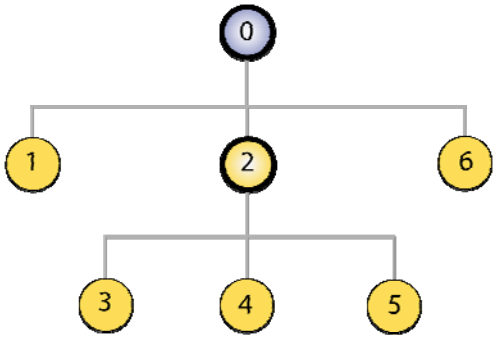
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied then exit Post Condition Rule: If satisfied, then previous Rollup Rules: Satisfied if at least 50% satisfied or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Set Activity 3's cmi.score.scaled to `0.75`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-04ab

Activity Tree Structure;



Sequencing Information;

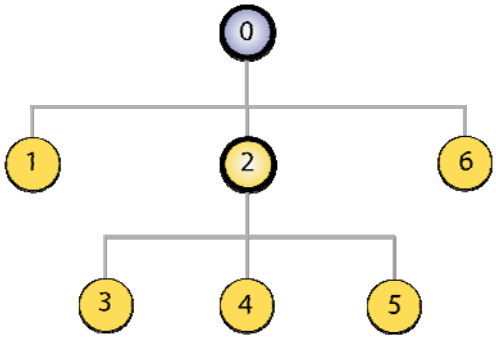
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied then exit Post Condition Rule: If satisfied, then previous Rollup Rules: Satisfied if at least 50% satisfied or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Set Activity 3's cmi.score.scaled to `0.75`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: RU-04ba

Activity Tree Structure:



Sequencing Information:

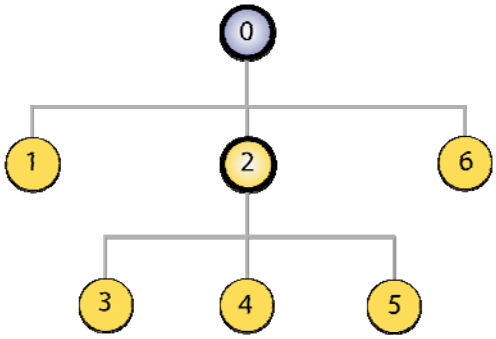
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if at least 50% completed or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a Start navigation request	Identify Activity 1 for delivery
2.	Process a Continue navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Process a Continue navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a Continue navigation request	Identify Activity 5 for delivery
5.	Set Activity 5s cmi.completion_status to `incomplete`; Process a Continue navigation request	Identify Activity 6 for delivery

Test Case: RU-04bb

Activity Tree Structure:



Sequencing Information:

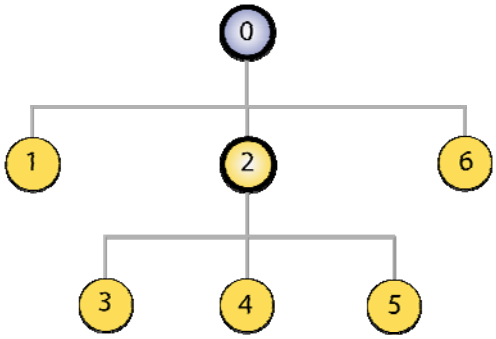
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if at least 50% completed or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Set Activity 3's cmi.score.scaled to `0.75`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Set Activity 4's cmi.score.scaled to `0.25`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-04bc

Activity Tree Structure:



Sequencing Information:

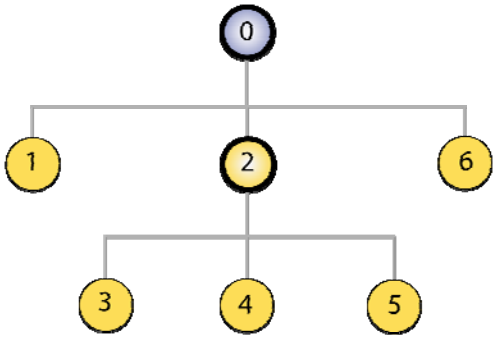
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if at least 50% completed or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5s cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 5 for delivery
7.	Process a *Previous* navigation request	Identify Activity 4 for delivery
8.	Set Activity 4's cmi.score.scaled to `0.25`; Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: RU-04bd

Activity Tree Structure:



Sequencing Information:

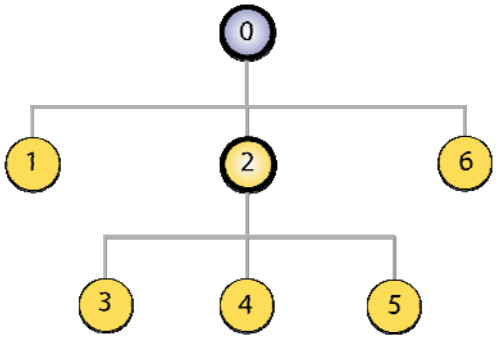
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if at least 50% completed or objective measure known
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5s cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 5 for delivery
7.	Set Activity 5s cmi.completion_status to `incomplete`; Process a *Previous* navigation request	Identify Activity 4 for delivery
8.	Set Activity 4's cmi.score.scaled to `0.8`; Process a *Continue* navigation request	Identify Activity 5 for delivery
9.	Set Activity 5's cmi.score.scaled to `0.5`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-05a

Activity Tree Structure:



Sequencing Information:

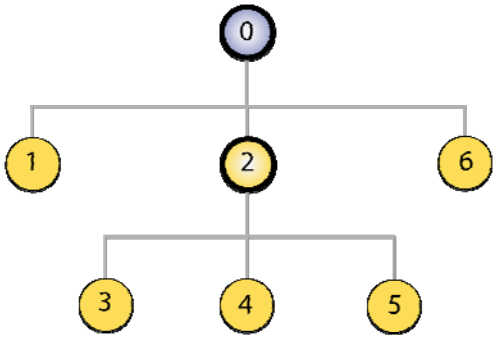
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if none satisfied
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-05b

Activity Tree Structure:



Sequencing Information:

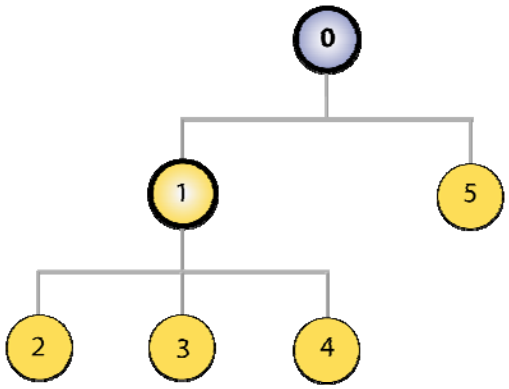
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then continue Rollup Rule: Satisfied if none completed
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.completion_status to `incomplete`; Process a *Previous* navigation request	Identify Activity 4 for delivery
6.	Set Activity 4's cmi.completion_status to `incomplete`; Process a *Previous* navigation request	Identify Activity 6 for delivery

Test Case: RU-06a

Activity Tree Structure:



Sequencing Information:

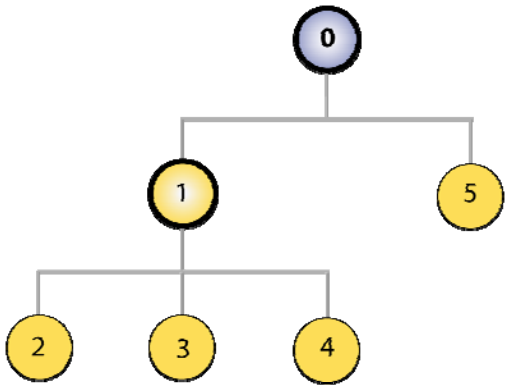
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true Choice == false Forward Only == true Use Current Attempt Objective Information == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then continue
2	Default
3	Default
4	Default
5	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Set Activity 2's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Process a *Previous* navigation request	Identify Activity 2 for delivery
6.	Set Activity 2's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery

Test Case: RU-06b

Activity Tree Structure:



Sequencing Information:

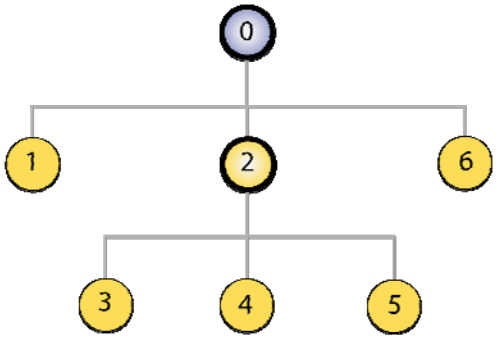
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true Choice == false Use Current Attempt Progress Information == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then continue Rollup Rules: Satisfied if at least 2 completed
2	Default
3	Default
4	Default
5	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Set Activity 2's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Process a *Previous* navigation request	Identify Activity 4 for delivery
6.	Set Activity 4's cmi.completion_status to `completed`; Process a *previous* navigation request	Identify Activity 5 for delivery

Test Case: RU-07a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If not satisfied, then retry
3	Limit Conditions: Attempt limit == 1 Sequencing Rules: Pre Condition Rule: If attempt limit exceeded, then skip Rollup Considerations: Required for Satisfied if not skipped Required for Not Satisfied if not skipped Rollup Rules: Rollup Progress Completion == false
4	Sequencing Rules: Pre Condition Rule: If always, then skip; Rollup Considerations: Required for Completion if attempted Required for Satisfied if attempted Required for Not Satisfied if attempted

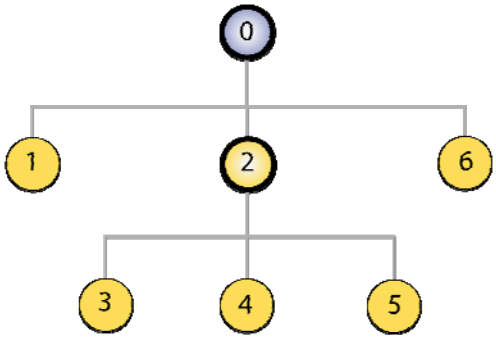
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
6.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-07b

Activity Tree Structure:



Sequencing Information:

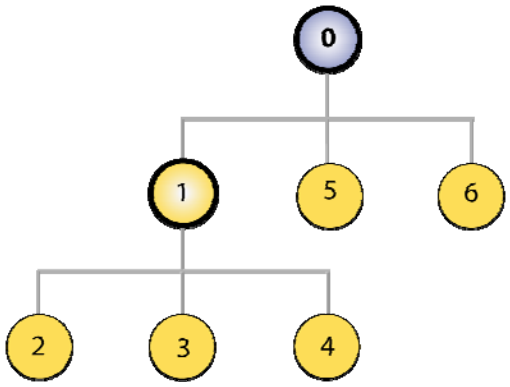
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If not satisfied, then retry
3	Limit Conditions: Attempt limit == 1 Sequencing Rules: Pre Condition Rule: If attempt limit exceeded, then skip Rollup Considerations: Required for Satisfied if not skipped Rollup Rules: Rollup Progress Completion == false
4	Sequencing Rules: Pre Condition Rule: If always, then skip Rollup Considerations: Required for Completion if attempted Required for Satisfied if attempted
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-07c

Activity Tree Structure:



Sequencing Information:

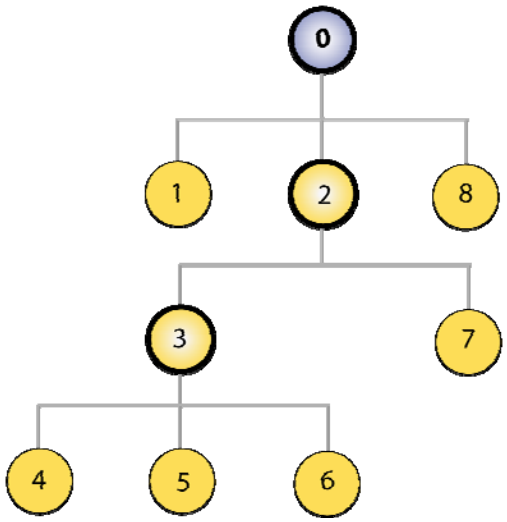
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If not satisfied, then retry
1	Control Mode: Flow == true Choice == false Limit Conditions: Attempt limit == 2 Sequencing Rules: Pre Condition Rule: If attempt limit exceeded, then skip Rollup Considerations: Required for Satisfied if not skipped Rollup Rules: Rollup Progress Completion == false
2	Default
3	Default
4	Default
5	Default
6	Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
6.	Set Activity 2's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery
7.	Process a *Continue* navigation request	Identify Activity 4 for delivery
8.	Process a *Continue* navigation request	Identify Activity 5 for delivery
9.	Set Activity 5's cmi.success_status to `failed`; Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery

Test Case: RU-08a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Post Condition Rule: If completed, then previous Rollup Rules: Completed if all Activity Progress Known and not completed
3	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then exit parent Rollup Rules: Incomplete if all Activity Progress Known or Objective Measure Known or Objective Status Known

Activity	Sequencing Information
	Satisfied if all attempted
4	Default
5	Default
6	Default
7	Rollup Considerations: Required for Completion if attempted
8	Default

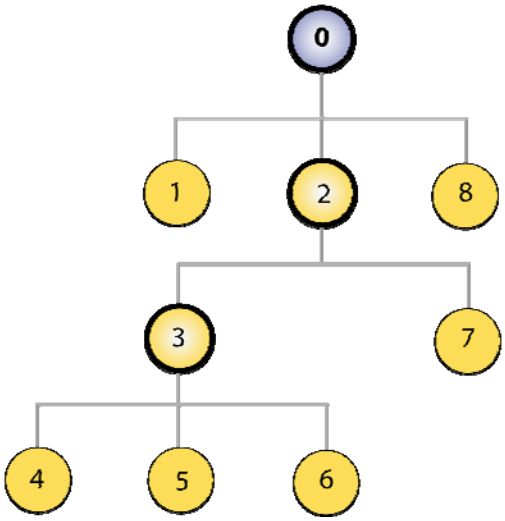
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 4 for delivery
3.	Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.score.scaled to `-0.5;` Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Set Activity 6's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-08b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Post Condition Rule: If completed, then previous Rollup Rules: Completed if all Activity Progress Known and not completed
3	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then exit parent Rollup Rules: Incomplete if all Activity Progress Known or Objective Measure Known or Objective Status Known Satisfied if all attempted

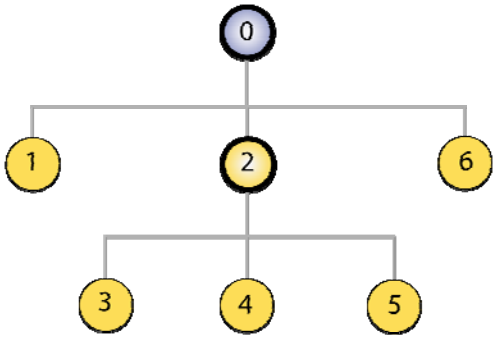
Activity	Sequencing Information
4	Delivery Controls: Objective Set by Content == true Completion Set by Content == true
5	Delivery Controls: Objective Set by Content == true Completion Set by Content == true
6	Delivery Controls: Objective Set by Content == true Completion Set by Content == true
7	Default
8	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 4 for delivery
3.	Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Process a *Continue* navigation request	Identify Activity 8 for delivery

Test Case: RU-09

Activity Tree Structure:



Sequencing Information:

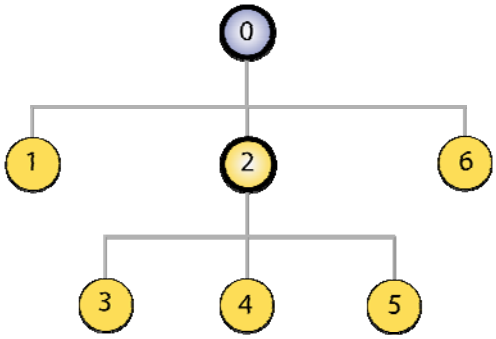
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If not completed, then retry all If completed, then continue
3	Default
4	Limit Conditions: Attempt limit == 2 Sequencing Rules: Pre Condition Rule: If attempt limit exceeded, then skip Rollup Considerations: Required for Incomplete if not skipped Required for Completed if not skipped Required for Satisfied if not skipped
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery
6.	Process a *Continue* navigation request	Identify Activity 3 for delivery
7.	Set Activity 3's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 4 for delivery
8.	Set Activity 4's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
9.	Set Activity 5's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery
10.	Process a *Continue* navigation request	Identify Activity 3 for delivery
11.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
12.	Set Activity 5's cmi.completion_status to `completed`; Process a *Previous* navigation request	Identify Activity 6 for delivery

Test Case: RU-10

Activity Tree Structure:



Sequencing Information:

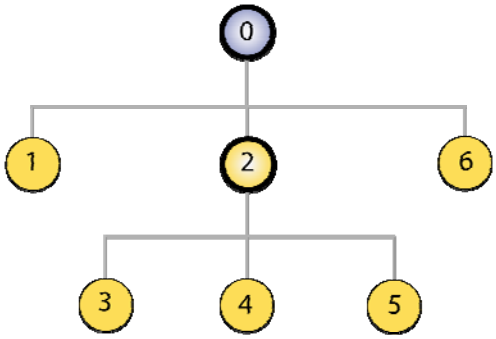
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if all satisfied Delivery Controls: Tracked == false
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery

Test Case: RU-11

Activity Tree Structure:



Sequencing Information:

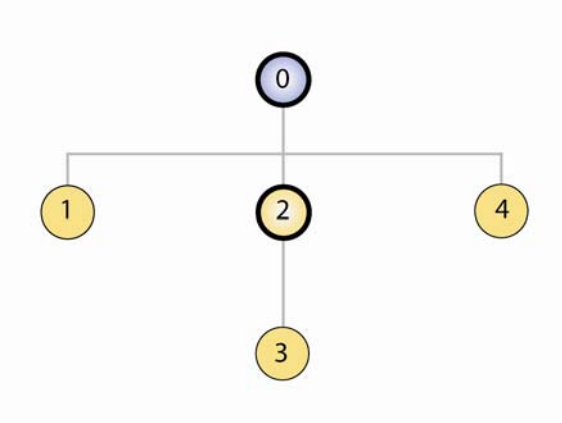
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if all not objective status known
3	Objectives: Primary Objective: *empty* Delivery Controls: Tracked == false
4	Delivery Controls: Tracked == false
5	Delivery Controls: Tracked == false
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: RU-12a

Activity Tree Structure:



Sequencing Information:

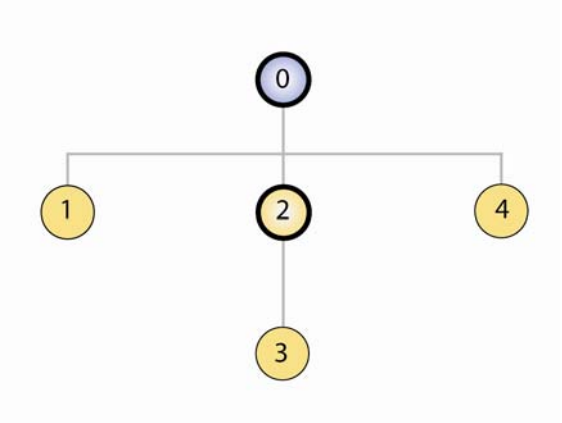
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If always, then exit Post Condition Rule: If not Objective Status Known, then previous Post Condition Rule: If satisfied, then retry Rollup Rules: Not satisfied if all not satisfied
3	Default
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Set Activity 3's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Previous* navigation request	Identify Activity 3 for delivery
5.	Set Activity 3's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
6.	Process Exit	End sequencing session.

Test Case: RU-12b

Activity Tree Structure:



Sequencing Information:

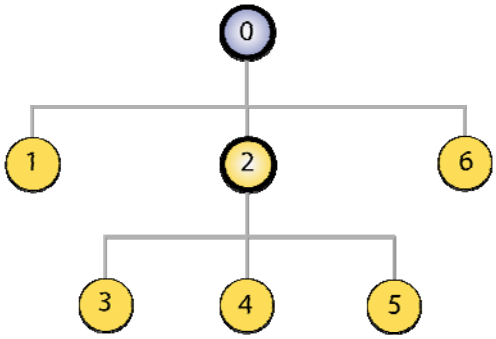
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If always, then exit Post Condition Rule: If not Activity Progress Known, then previous Post Condition Rule: If completed, then retry Rollup Rules: Incomplete if all incomplete
3	Default
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Set Activity 3's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Previous* navigation request	Identify Activity 3 for delivery
5.	Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
6.	Process Exit	End sequencing session.

Test Case: MS-01

Activity Tree Structure:



Sequencing Information:

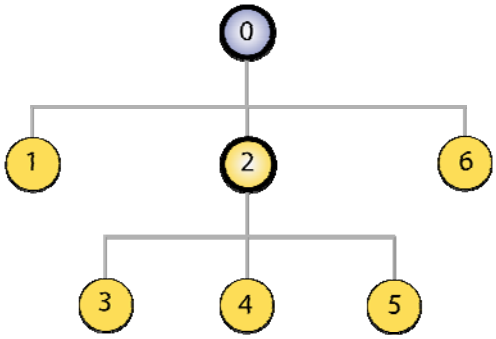
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rule: Pre Condition Rule: If objective measure greater than 0.4, then skip
3	Rollup Rules: Rollup Objective Satisfied == false
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: MS-02

Activity Tree Structure:



Sequencing Information:

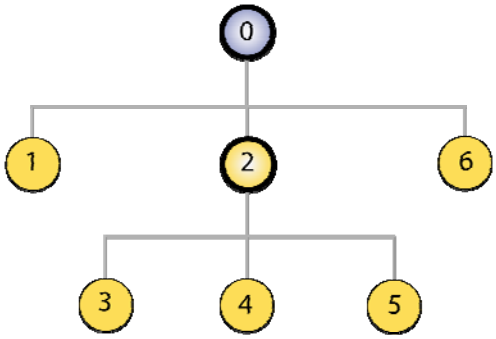
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rule: Pre Condition Rule: If objective measure less than 0.6, then skip
3	Rollup Rules: Rollup Objective Measure Weight == 0.75
4	Rollup Rules: Rollup Objective Measure Weight == 0.25
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `0.25`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `0.5`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: MS-03

Activity Tree Structure:



Sequencing Information:

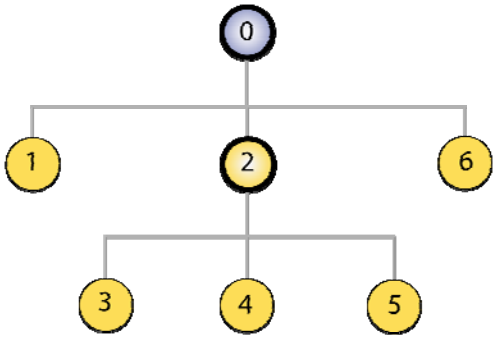
Activity	Sequencing Information
0	Control Mode; Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If objective measure greater than 0.50, then skip
3	Rollup Rules: Rollup Objective Measure Weight == 0.50
4	Rollup Rules: Rollup Objective Measure Weight == 0.00
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `-1.0`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `0.5`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: MS-04

Activity Tree Structure:



Sequencing Information:

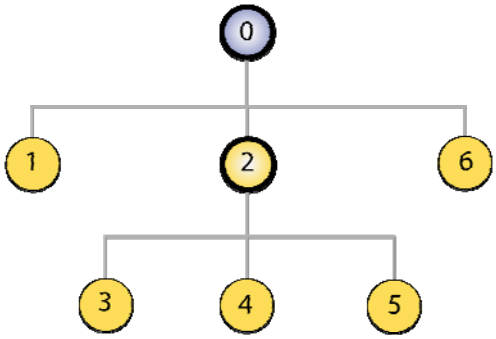
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If objective measure less than 0.25, then skip
3	Default
4	Delivery Controls: Tracked == false
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `0.1`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `0.5`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: MS-05a

Activity Tree Structure:



Sequencing Information:

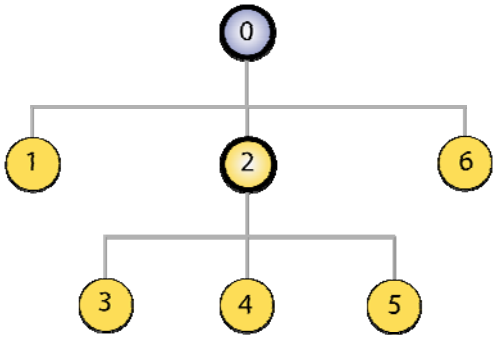
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Objectives: Primary Objective: Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then previous
3	Default
4	Default
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: MS-05b

Activity Tree Structure:



Sequencing Information:

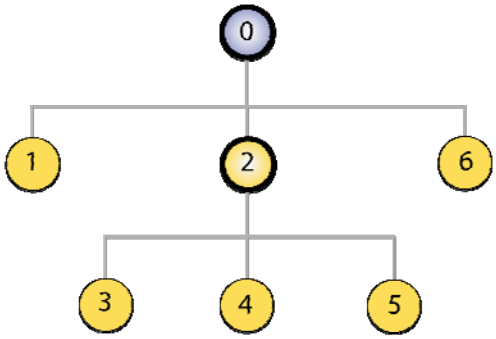
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Objectives: Primary Objective: ObjectiveID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then previous
3	Default
4	Default
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `0.5`; Set Activity 3's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `0.75`; Set Activity 4's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `1.0`; Set Activity 5's cmi.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: MS-06

Activity Tree Structure:



Sequencing Information:

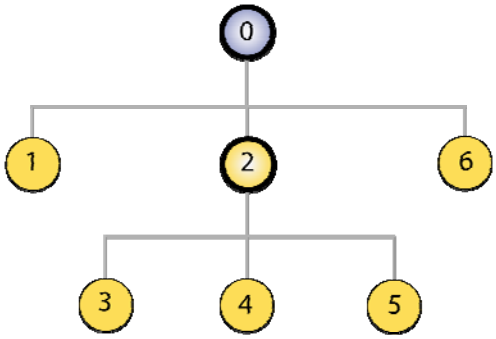
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Objectives: Primary Objective: Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then previous Pre Condition Rule: If satisfied, then skip Rollup Considerations: Measure Satisfaction If Active == false
3	Default
4	Default
5	Rollup Rules: Rollup Objective Measure Weight == 0.25
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.score.scaled to `0.75`; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.score.scaled to `0.5`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: SX-02

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If satisfied, then exit Post Condition Rule: If satisfied, then previous Rollup Rules: Satisfied if all completed
3	Completion Threshold** == 0.5
4	Default
5	Completion Threshold** == 0.75
6	Default

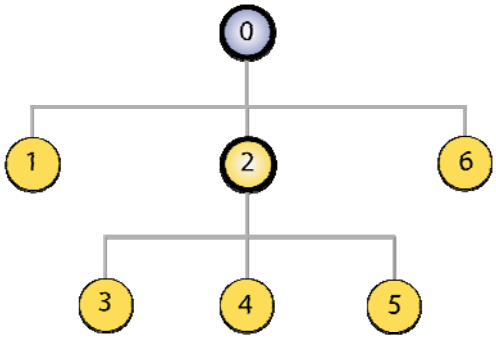
** This element is defined in the http://www.adlnet.org/xsd/adlcp_v1p3 namespace.

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.completion_status to `incomplete`; Set Activity 3's cmi.progress_measure to `0.5`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Set Activity 4's cmi.success_status to failed; Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.progress_measure to `0.9`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: SX-03

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-SX03-1 Read Satisfied Status == false Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-SX03-2 Read Satisfied Status == false Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-SX03-3 Read Satisfied Status == false Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true

Activity	Sequencing Information
2	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then previous Rollup Rules: Completed if all satisfied
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-SX03-1
4	Default
5	Default
6	Default

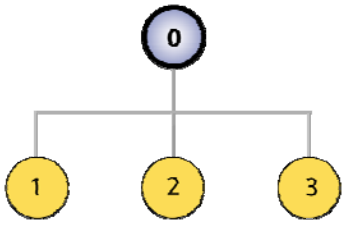
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's score.scaled to `0.75`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 1's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.score.scaled to `0.90`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 1's cmi.objectives.n with ID of `obj3`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: SX-04a

Activity Tree Structure:



Sequencing Information:

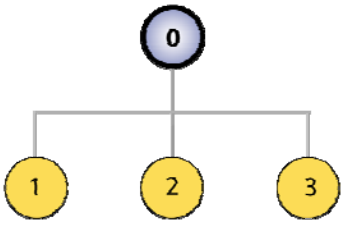
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-SX04a Write Satisfied Status == true Read Satisfied Status == true
2	Default
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process an *Abandon All* navigation request	Abandon the Activity Tree and End the Sequencing Session
3.	Process a *Start* navigation request	Identify Activity 1 for delivery

Test Case: SX-04b

Activity Tree Structure:



Sequencing Information:

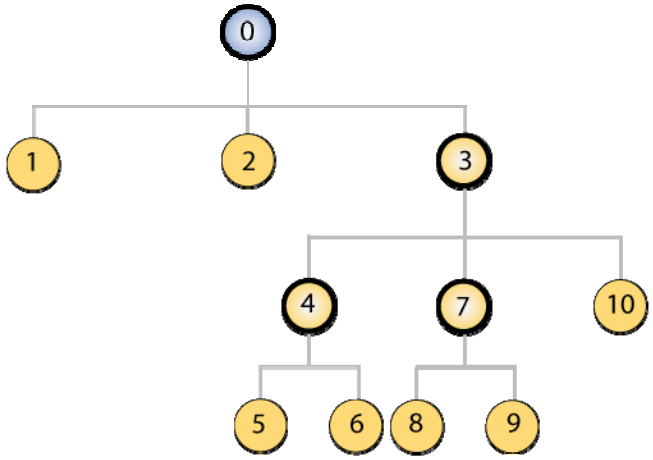
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == true
1	Sequencing Rules: Pre Condition Rule: If completed or satisfied, then skip
2	Default
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process an *Abandon* navigation request	Abandon Activity 1
3.	Process a *Choice* navigation request for Activity 0	Identify Activity 1 for delivery

Test Case: SX-05

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-SX05-1 Read Normalized Measure == false Write Satisfied Status == true
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false Objectives: Primary Objective: Objective ID == PRIMARYOBJ

Activity	Sequencing Information
	Map Info: Target Objective ID == gObj-SX05-2 Read Normalized Measure == false Write Satisfied Status == true
3	Control Mode: Flow == true Choice == false Sequencing Rules: Post Condition Rule: If obj-SX05-5 not satisfied, then retry Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-5 Map Info: Target Objective ID == gObj-SX05-5 Read Normalized Measure == false Write Satisfied Status == true
4	Control Mode: Flow == true Choice == false
5	Sequencing Rules: Pre Condition Rule: If obj-SX05-3a satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-3a Map Info: Target Objective ID == gObj-SX05-3a Read Normalized Measure == false Write Satisfied Status == true
6	Sequencing Rules: Pre Condition Rule: If obj-SX05-3b satisfied, then skip Post Condition Rule: If obj-SX05-3a not satisfied or obj-SX05-3b not satisfied, then retry all Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-3a Map Info: Target Objective ID == gObj-SX05-3a Read Normalized Measure == false Write Satisfied Status == true Objective: Objective ID == obj-SX05-3b

Activity	Sequencing Information
	Map Info: Target Objective ID == gObj-SX05-3b Read Normalized Measure == false Write Satisfied Status == true
7	Control Mode: Flow == true Choice == false
8	Sequencing Rules: Pre Condition Rule: If obj-SX05-4a satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-4a Map Info: Target Objective ID == gObj-SX05-4a Read Normalized Measure == false Write Satisfied Status == true
9	Sequencing Rules: Pre Condition Rule: If obj-SX05-4b satisfied, then skip Post Condition Rule: If obj-SX05-4a not satisfied or obj-SX05-4b not satisfied, then retry all Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-4a Map Info: Target Objective ID == gObj-SX05-4a Read Normalized Measure == false Write Satisfied Status == true Objective: Objective ID == obj-SX05-4b Map Info: Target Objective ID == gObj-SX05-4b Read Normalized Measure == false Write Satisfied Status == true
10	Sequencing Rules: Post Condition Rule: If always, then exit parent Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj-SX05-3a Map Info: Target Objective ID == gObj-SX05-3a Read Normalized Measure == false

Activity	Sequencing Information
	Write Satisfied Status == true Objective: Objective ID == obj-SX05-3b Map Info: Target Objective ID == gObj-SX05-3b Read Normalized Measure == false Write Satisfied Status == true Objective: Objective ID == obj-SX05-4a Map Info: Target Objective ID == gObj-SX05-4a Read Normalized Measure == false Write Satisfied Status == true Objective: Objective ID == obj-SX05-4b Map Info: Target Objective ID == gObj-SX05-4b Read Normalized Measure == false Write Satisfied Status == true Objective: Objective ID == obj-SX05-5 Map Info: Target Objective ID == gObj-SX05-5 Read Normalized Measure == false Write Satisfied Status == true

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set Activity 2's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Find the index of Activity 5's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 8 for delivery
6.	Find the index of Activity 8's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 9 for delivery
7.	Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 10 for delivery
8.	Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID	Identify Activity 5 for delivery

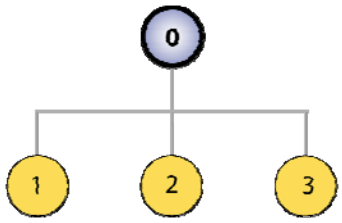
	of `obj-SX05-5`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	
9.	Find the index of Activity 5's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery
10.	Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 6 for delivery
11.	Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 8 for delivery
12.	Find the index of Activity 8's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 9 for delivery
13.	Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 9 for delivery
14.	Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 10 for delivery
15.	Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`;	Identify Activity 6 for delivery

	Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-5`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	
16.	Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 6's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 9 for delivery
17.	Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 9's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 10 for delivery
18.	Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3a`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-3b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4a`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-4b`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 10's cmi.objectives.n with ID of `obj-SX05-5`; Set that objective's cmi.objectives.n.success_status to	End Sequencing Session

	`passed`; Process a *Continue* navigation request	

Test Case: SX-06

Activity Tree Structure:



Sequencing Information:

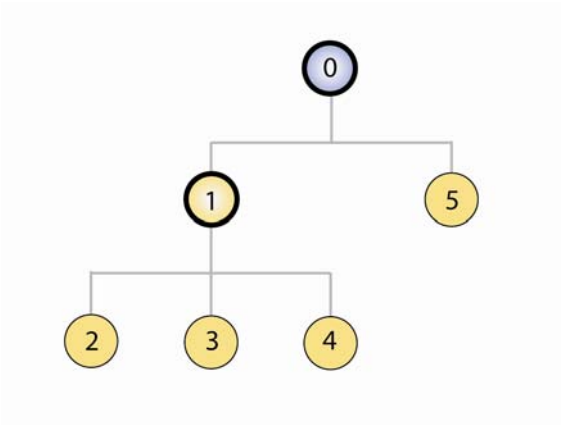
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Sequencing Rules: Post Condition Rule: If completed, then continue Post Condition Rule: If not completed, then retry
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set Activity 2's cmi.completion_status to `not attempted`; Process a *Continue* navigation request	Identify Activity 2 for delivery
4.	Set Activity 2's cmi.completion_status to `completed`; Process a *Previous* navigation request	Identify Activity 3 for delivery

Test Case: SX-07a

Activity Tree Structure:



Sequencing Information:

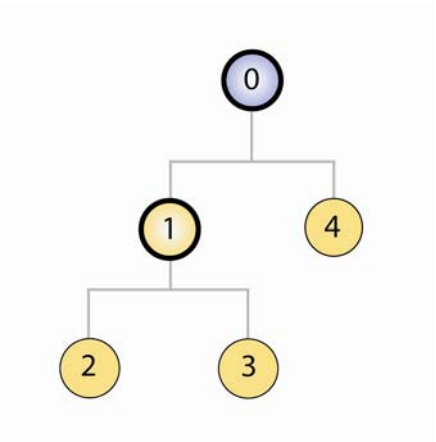
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true Choice == false Forward Only == true
2	Sequencing Rules: Pre Condition Rule: If not completed, then skip
3	Default
4	Default
5	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Set Activity 2's cmi.completion_status to `incomplete`; Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Process a *Previous* navigation request	Identify Activity 3 for delivery
6.	Process *Exit All* navigation request	End sequencing session.

Test Case: SX-07b

Activity Tree Structure:



Sequencing Information:

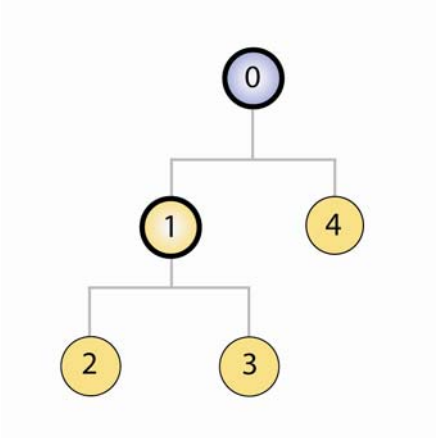
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == true
1	Control Mode: Flow == true Choice == false
2	Sequencing Rules: Pre Condition Rule: If not satisfied, then skip
3	Default
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Set Activity 2's cmi.success_status to `failed` Process a *Choice* navigation request for Activity 1	Identify Activity 3 for delivery
3.	Process *Exit All* navigation request	End sequencing session.

Test Case: SX-07c

Activity Tree Structure:



Sequencing Information:

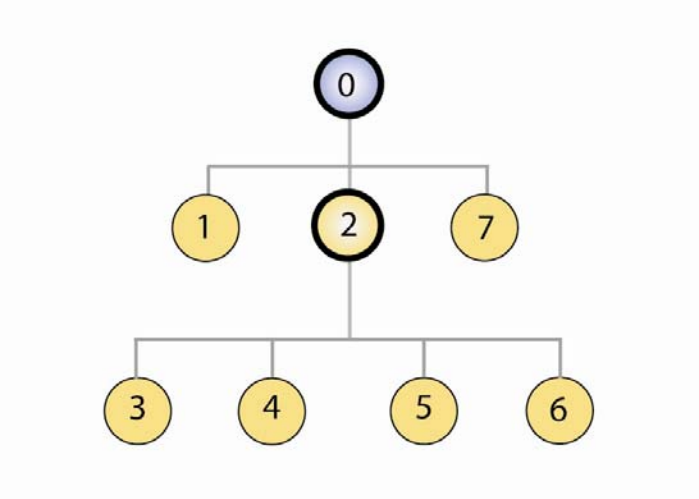
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If completed, then retry
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip
3	Default
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Process *Exit All* navigation request	End sequencing session.

Test Case: SX-07d

Activity Tree Structure:



Sequencing Information:

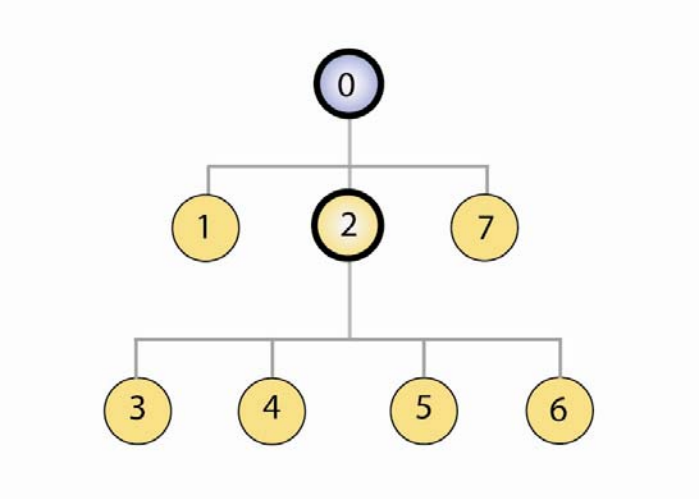
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false
3	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true; Minimum Normalized Measure == -0.50
4	Sequencing Rules: Pre Condition Rule: If completed, then skip
5	Sequencing Rules: Pre Condition Rule: If satisfied, then skip
6	Default
7	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Set Activity 3's cmi.success_status to `failed`; Set Activity 3's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Process a *Continue* navigation request	Identify Activity 7 for delivery
7.	Process a *Previous* navigation request	Identify Activity 6 for delivery
8	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: SX-07e

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Control Mode: Flow == true Choice == false Forward Only == true
3	Default
4	Sequencing Rules: Pre Condition Rule: If satisfied, then skip
5	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true; Minimum Normalized Measure == 0.0
6	Sequencing Rules: Pre Condition Rule: If completed, then skip

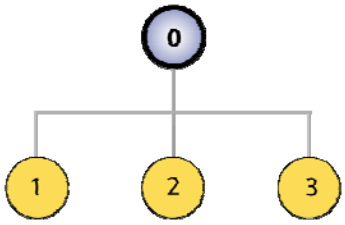
7	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 3 for delivery
3.	Process a *Continue* navigation request	Identify Activity 4 for delivery
4.	Process a *Continue* navigation request	Identify Activity 5 for delivery
5.	Set Activity 5's cmi.success_status to `failed`; Set Activity 5's cmi.score.scaled to `0.90`; Process a *Continue* navigation request	Identify Activity 6 for delivery
6.	Set Activity 6's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 7 for delivery
7.	Process a *Previous* navigation request	Identify Activity 3 for delivery
8	Process a *Continue* navigation request	Identify Activity 7 for delivery

Test Case: OB-01a

Activity Tree Structure:



Sequencing Information:

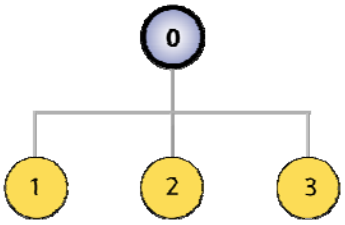
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: ID == PRIMARYOBJ_1 Map Info: Target Objective ID == gObj-OB01a Write Satisfied Status == true
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: ID == PRIMARYOBJ_2 Map Info: Target Objective ID == gObj-OB01a
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-01b

Activity Tree Structure:



Sequencing Information:

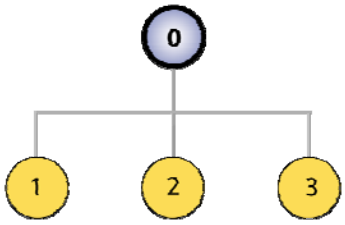
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB01b Write Normalized Measure == true
2	Sequencing Rules: Pre Condition Rule: If objective measure greater than 0.75, then skip Objectives: Primary Objective: ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB01b
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `0.8`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-01c

Activity Tree Structure:



Sequencing Information:

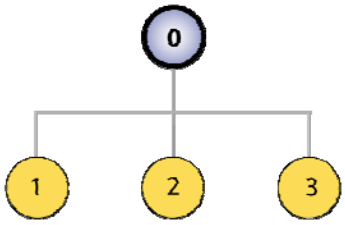
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB01c Write Satisfied Status == true Write Normalized Measure == true
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true; Minimum Normalized Measure == 0.50 Map Info: Target Objective ID == gObj-OB01c
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `failed`; Set Activity 1's cmi.score.scaled to `0.8`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-02a

Activity Tree Structure:



Sequencing Information:

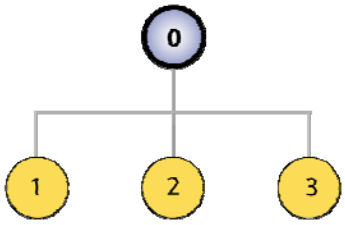
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB02a Write Satisfied Status == true
2	Sequencing Rules: Pre Condition Rule: If obj1 not satisfied, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB02a
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-02b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB02b Write Normalized Measure == true
2	Sequencing Rule: Pre Condition Rule: If obj1 objective measure less than 0.25, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB02b
3	Default

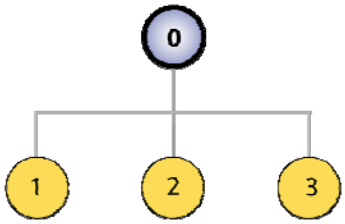
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-03a

NOTE: Test Cases OB-03a and OB-03b and OB-03c test cross-activity tree persistence of global shared objectives; they should be performed back to back – OB-03a first, followed by OB-03b, and followed by OB-03 c.

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB03-1 Write Normalized Measure == true Write Satisfied Status == true
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB03-2 Write Normalized Measure == true Write Satisfied Status == true
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB03-3

	Write Normalized Measure == true Write Satisfied Status == true

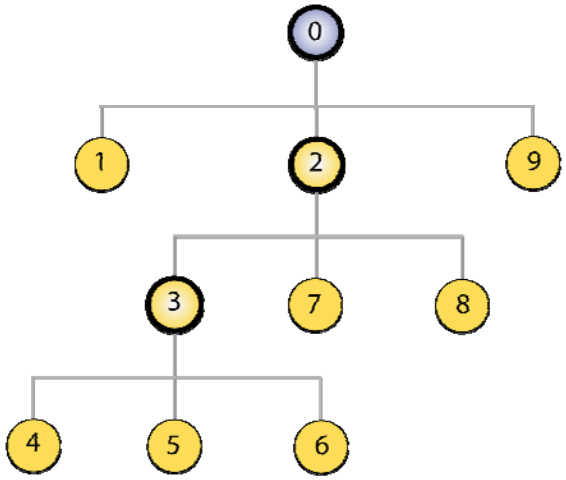
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.0`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.5`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.75`; Process a *Exit All* navigation request	End Sequencing Session

Test Case: OB-03b

NOTE: Test Cases OB-03a and OB-03b and OB-03c test cross-activity tree persistence of global shared objectives; they should be performed backed to back – OB-03a first, followed by OB-03b, and followed by OB-03c.

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0 **	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-2 Sequencing Rules: Pre Condition Rule: If not satisfied, then skip
2	Control Mode: Flow == true Choice == false
3	Control Mode: Flow == true Choice == true Objectives: Primary Objective: Objective ID == PRIMARYOBJ

Activity	Sequencing Information
	Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-OB03-3 Read Satisfied Status == false Write Satisfied Status == true Write Normalized Measure == true Sequencing Rules: Pre Condition Rule: If satisfied, then skip
4	Default
5	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-1 Sequencing Rules: Pre Condition Rule: If satisfied, then skip
6	Default
7	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-3 Sequencing Rules: Pre Condition Rule: If satisfied, then skip
8	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-3 Sequencing Rules: Pre Condition Rule: If satisfied, then skip
9	Default

** This activity (the root of the activity tree) has Objectives Global to System == false.

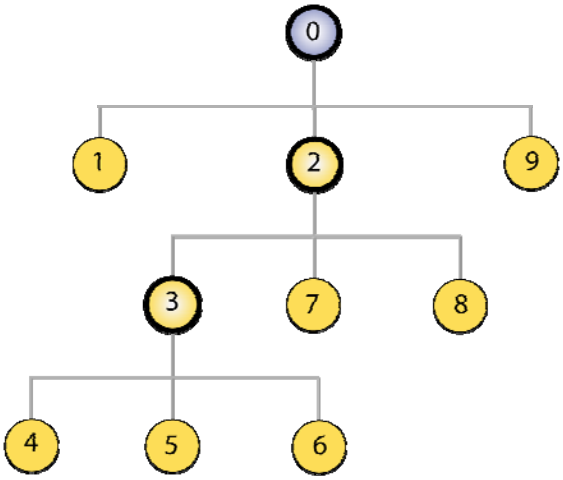
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 4 for delivery
3.	Set Activity 4's cmi.score.scaled to 1.0; Process a *Continue* navigation request	Identify Activity 5 for delivery
4.	Set Activity 5's cmi.score.scaled to 1.0; Process a *Continue* navigation request	Identify Activity 6 for delivery
5.	Set Activity 6's cmi.score.scaled to 1.0; Set Activity 6's cmi.exit to `suspend`; Process a *Exit All* navigation request	End sequencing session
6.	Process a *Start* navigation request	Identify Activity 1 for delivery
7.	Process a *Continue* navigation request	Identify Activity 9 for delivery
8.	Process a *Choice* navigation request for Activity 6	Identify Activity 6 for delivery
9.	Process an *Exit All* navigation request	End sequencing session
10.	Process a *Start* navigation request	Identify Activity 1 for delivery
11.	Process a *Continue* navigation request	Identify Activity 4 for delivery
12.	Set Activity 4's cmi.score.scaled to 1.0; Process a *Continue* navigation request	Identify Activity 5 for delivery
13.	Set Activity 5's cmi.score.scaled to 1.0; Process a *Continue* navigation request	Identify Activity 6 for delivery
14.	Set Activity 6's cmi.score.scaled to 1.0; Process a *Continue* navigation request	Identify Activity 9 for delivery

Test Case: OB-03c

NOTE: Test Cases OB-03a and OB-03b and OB-03c test cross-activity tree persistence of global shared objectives; they should be performed backed to back – OB-03a first, followed by OB-03b, and followed by OB-03 c.

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-3 Sequencing Rules: Pre Condition Rule: If objective status not known, then skip
2	Control Mode: Flow == true Choice == false
3	Control Mode: Flow == true Choice == false Objectives: Primary Objective: Objective ID == PRIMARYOBJ

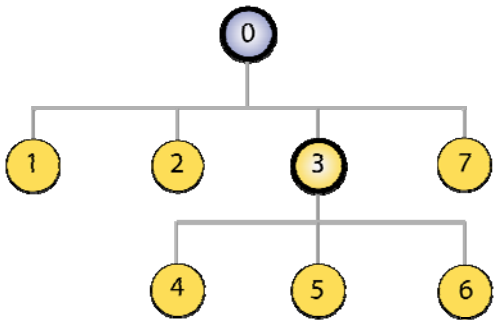
Activity	Sequencing Information
	Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-OB03-3 Read Satisfied Status == false Write Satisfied Status == true Sequencing Rules: Pre Condition Rule: If satisfied, then skip
4	Default
5	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-1 Sequencing Rules: Pre Condition Rule: If satisfied, then skip
6	Default
7	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-2 Sequencing Rules: Pre Condition Rule: If not satisfied, then skip
8	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB03-2 Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB03-3 Sequencing Rules: Pre Condition Rule: If not satisfied and obj1 objective status not known, then skip
9	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 9 for delivery

Test Case: OB-04

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB04-1 Read Satisfied Status == false Read Normalized Measure == false Write Satisfied Status == true Map Info: Target Objective ID == gObj-OB04-3 Read Satisfied Status == false Read Normalized Measure == false Write Normalized Measure == true
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB04-2 Write Satisfied Status == true
3	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If completed, then skip Rollup Rules: Completed if all Satisfied

Activity	Sequencing Information
4	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB04-1
5	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB04-2
6	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == -0.75 Map Info: Target Objective ID == gObj-OB04-3
7	Default

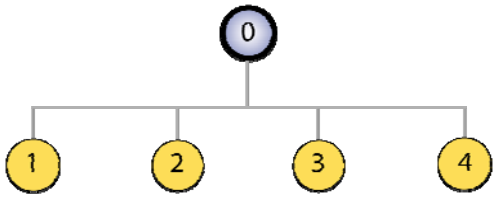
Test Script:

This test contains several questions in which the tester must answer dealing with the availability of user interface devices provided by the LMS.

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `-0.25`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set Activity 2's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 7 for delivery

Test Case: OB-05a

Activity Tree Structure:



Sequencing Information:

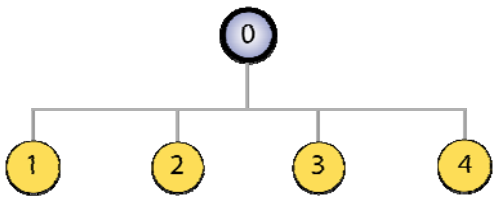
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Map Info: Target Objective ID == gObj-OB05a Write Satisfied Status == true Write Normalized Measure == true
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.4 Map Info: Target Objective ID == gObj-OB05a Sequencing Rules: Pre Condition Rule: If satisfied, then skip
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Minimum Normalized Measure == 0.1 Map Info: Target Objective ID == gObj-OB05a Sequencing Rules: Pre Condition Rule: If not satisfied, then skip
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `0.85`; Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery

Test Case: OB-05b

Activity Tree Structure:



Sequencing Information:

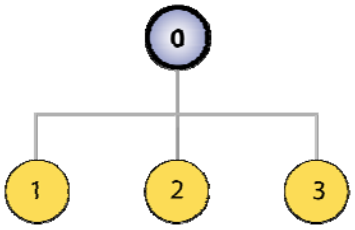
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB05b Write Satisfied Status == true Write Normalized Measure == true
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.75 Map Info: Target Objective ID == gObj-OB05b Sequencing Rules: Pre Condition Rule: If not objective status known , then skip
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Minimum Normalized Measure == 0.1 Map Info: Target Objective ID == gObj-OB05b Sequencing Rules: Pre Condition Rule: If satisfied, then skip
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 4 for delivery

Test Case: OB-05c

Activity Tree Structure:



Sequencing Information:

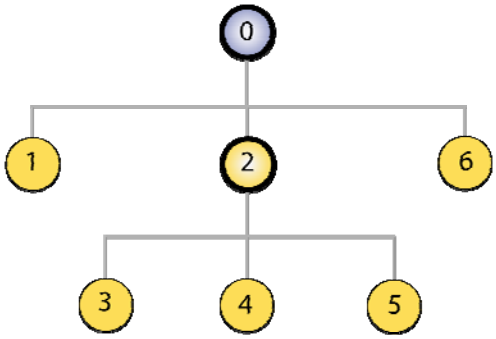
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.0 Map Info: Target Objective ID == gObj-OB05c Write Satisfied Status == true Write Normalized Measure == true
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB05c Sequencing Rules: Pre Condition Rule: If not objective status known , then skip
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-06

Activity Tree Structure:



Sequencing Information:

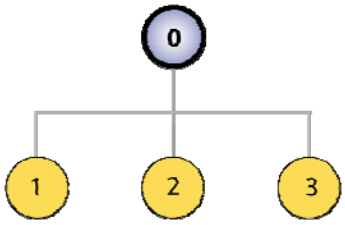
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB06 Write Satisfied Status == true Write Normalized Measure == true
2	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If objective measure known and objective status known, then skip Pre Condition Rule: If objective measure known or objective status known, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB06 Delivery Controls: Tracked == false
3	Default
4	Default
5	Default
6	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Set Activity 1's cmi.score.scaled to `0.50`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-07a

Activity Tree Structure:



Sequencing Information:

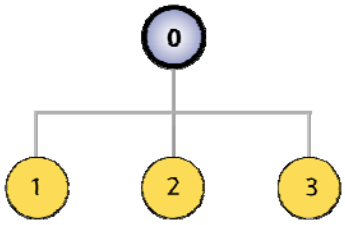
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07a Write Satisfied Status == true
2	Sequencing Rule: Pre Condition Rule: If obj1 Objective Status Known and not satisfied, then skip Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07a
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07a

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-07b

Activity Tree Structure:



Sequencing Information:

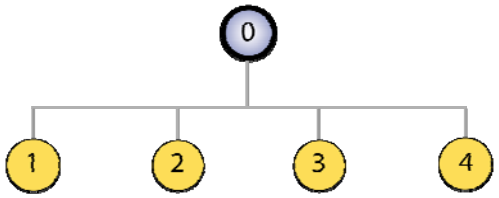
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07b Write Normalized Measure == true
2	Sequencing Rule: Pre Condition Rule: If obj1 objective measure less than 0.75, then skip Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07b
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB07b

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.7`; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-08a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08 Write Normalized Measure == true
2	Sequencing Rule: Pre Condition Rule: If measure greater than 0.0, then skip Post Condition Rule: If always, then exit parent Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08
3	Sequencing Rule: Pre Condition Rule: If measure greater than 0.5 or less than 0.5, then skip Post Condition Rule: If always, then exit parent Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08
4	Sequencing Rule: Pre Condition Rule: If less than 1.0, then skip Post Condition Rule: If always, then exit parent Objectives:

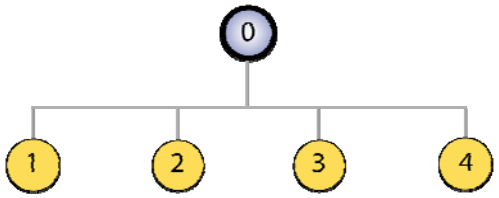
	Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 4 for delivery
3.	Process a *Continue* navigation request	End Sequencing Session

Test Case: OB-08b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08 Write Normalized Measure == true
2	Sequencing Rule: Pre Condition Rule: If measure greater than 0.0, then skip Post Condition Rule: If always, then exit parent Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08
3	Sequencing Rule: Pre Condition Rule: If measure greater than 0.5 or less than 0.5, then skip Post Condition Rule: If always, then exit parent Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08
4	Sequencing Rule: Pre Condition Rule: If less than 1.0, then skip Post Condition Rule: If always, then exit parent Objectives:

	Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB08

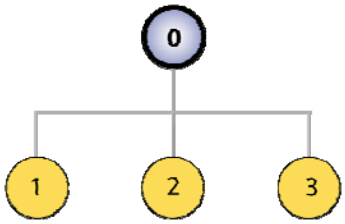
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	End Sequencing Session

Test Case: OB-09a

NOTE: Test Cases OB-09a and OB-09b test cross-activity tree persistence of global shared objectives; they should be performed back to back – OB-09a first, followed by OB-09b.

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If not satisfied, then retry
1	Sequencing Rule: Pre Condition Rule: If obj1 satisfied, then skip Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB09-obj1 Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false
2	Sequencing Rule: Pre Condition Rule: If obj2 satisfied, then skip Objectives: Primary Objective: Objective ID == obj2

	Map Info: Target Objective ID == gObj-OB09-obj2 Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB09-obj1 Write Satisfied Status == true Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-OB09-obj2 Write Satisfied Status == true

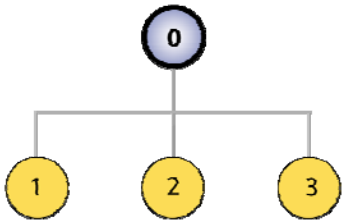
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 3's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.success_status to `passed`; Set Activity 3's cmi.success_status to `failed`; Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery
5.	Process a *Continue* navigation request	Identify Activity 3 for delivery
6.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 3's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.success_status to `passed`; Set Activity 3's cmi.success_status to `passed`; Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	End Sequencing Session

Test Case: OB-09b

NOTE: Test Cases OB-09a and OB-09b test cross-activity tree persistence of global shared objectives; they should be performed back to back – OB-09a first, followed by OB-09b.

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If not satisfied, then retry
1	Sequencing Rule: Pre Condition Rule: If obj1 satisfied, then skip Objectives: Primary Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB09-obj1 Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false
2	Sequencing Rule: Pre Condition Rule: If obj2 satisfied, then skip Objectives: Primary Objective: Objective ID == obj2

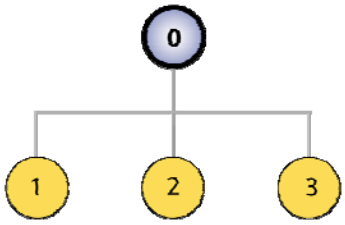
	Map Info: Target Objective ID == gObj-OB09-obj2 Rollup Rules: Rollup Progress Completion == false Rollup Objective Satisfied == false
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB09-obj1 Write Satisfied Status == true Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-OB09-obj2 Write Satisfied Status == true

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 3 for delivery
2.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 3's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.success_status to `failed`; Set Activity 3's cmi.success_status to `failed`; Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 3's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.success_status to `failed`; Set Activity 3's cmi.success_status to `failed`; Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	Identify Activity 1 for delivery
5.	Process a *Continue* navigation request	Identify Activity 2 for delivery
6.	Process a *Continue* navigation request	Identify Activity 3 for delivery
7.	Find the index of Activity 3's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 3's cmi.objectives.n with ID of `obj2`; Set that objective's cmi.objectives.n.success_status to `passed`; Set Activity 3's cmi.success_status to `passed`; Set Activity 3's cmi.completion_status to `completed`; Process a *Continue* navigation request	End Sequencing Session

Test Case: OB-10a

Activity Tree Structure:



Sequencing Information:

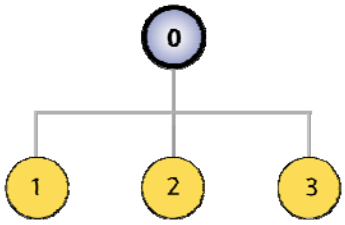
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10a Read Satisfied Status == false Write Satisfied Status == true Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10a Write Satisfied Status == true
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10a Delivery Controls: Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of PRIMARYOBJ; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of PRIMARYOBJ; Set that objective's cmi.objectives.n.success_status to failed; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-10b

Activity Tree Structure:



Sequencing Information:

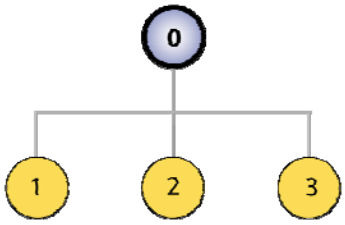
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10b Read Satisfied Status == false Write Normalized Measure == true Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If objective measure greater than 0.7, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10b Write Normalized Measure == true
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10b Delivery Controls: Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of PRIMARYOBJ; Set that objective's cmi.objectives.n.score.scaled to `0.7`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of PRIMARYOBJ; Set that objective's cmi.objectives.n.score.scaled to 0.4; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-10c

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10c Read Satisfied Status == false Write Satisfied Status == true Objective: Objective ID == obj1 Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10c Write Satisfied Status == true
3	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10c Delivery Controls:

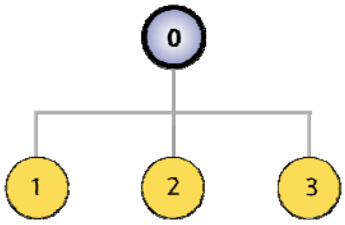
	Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set Activity 2's cmi.success_status to failed; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-10d

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10d Read Satisfied Status == false Write Normailized Measure == true Objective: Objective ID == obj1 Objective: Objective ID == obj2 Objective: Objective ID == obj3 Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If objective measure greater than 0.7, then skip Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10d Write Normalized Measure == true
3	Objectives: Primary Objective:

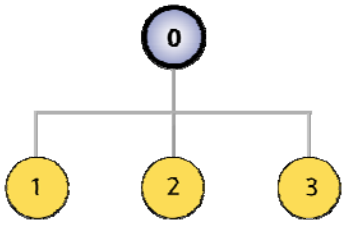
	Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB10d Delivery Controls: Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Set Activity 1's cmi.score.scaled to `0.7`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set Activity 2's cmi.score.scaled to 0.4; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-11a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj Map Info: Target Objective ID == gObj-OB11a Read Satisfied Status == false Write Satisfied Status == true Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If satisfied, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj Map Info: Target Objective ID == gObj-OB11a Write Satisfied Status == true
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj Map Info: Target Objective ID == gObj-OB11a

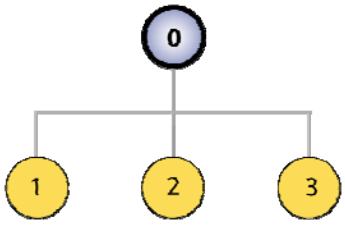
	Delivery Controls: Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of obj; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of obj; Set that objective's cmi.objectives.n.sucess_status to failed; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-11b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB11b Read Satisfied Status == false Write Satisfied Status == true Objective: Objective ID == obj2 Objective: Objective ID == obj3 Delivery Controls: Tracked == false
2	Sequencing Rules: Pre Condition Rule: If objective measure greater than 0.7, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj Map Info: Target Objective ID == gObj-OB11b Write Satisfied Status == true
3	Objectives: Primary Objective: *empty*

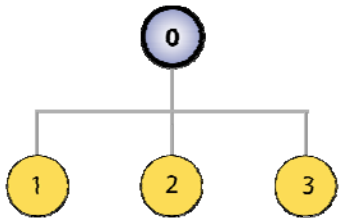
	Objective: Objective ID == obj Map Info: Target Objective ID == gObj-OB11b Delivery Controls: Tracked == false

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of obj; Set that objective's cmi.objectives.n.score.scaled to `0.7`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of obj; Set that objective's cmi.objectives.n.score.scaled to 0.4; Process a *Continue* navigation request	Identify Activity 3 for delivery

Test Case: OB-12a

Activity Tree Structure:



Sequencing Information:

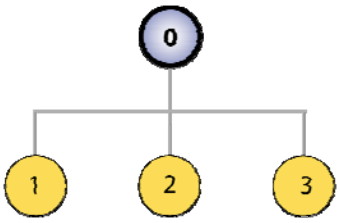
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Sequencing Rules: Post Condition Rule: If obj1 Objective Status known and Objective Measure not known, then previous Objectives: Primary Objective: *empty* Objective: Objective ID == obj1
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 1 for delivery

Test Case: OB-12b

Activity Tree Structure:



Sequencing Information:

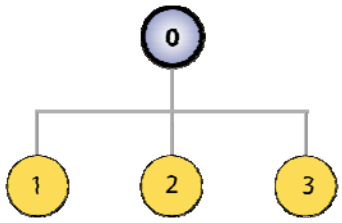
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Sequencing Rules: Post Condition Rule: If obj1 Objective Status not known and Objective Measure known, then continue Objectives: Primary Objective: *empty* Objective: Objective ID == obj1
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.4`; Process a *Previous* navigation request	Identify Activity 3 for delivery

Test Case: OB-12c

Activity Tree Structure:



Sequencing Information:

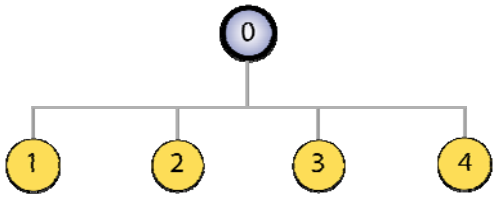
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Sequencing Rules: Post Condition Rule: If obj1 Objective Status known and Objective Measure known, then continue Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.9
3	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.4`; Process a *Previous* navigation request	Identify Activity 3 for delivery

Test Case: OB-13a

Activity Tree Structure:



Sequencing Information:

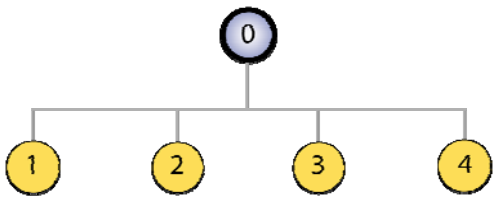
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB13a Write Satisfied Status == true Write Normalized Measure = true
3	Sequencing Rules: Pre Condition Rule: If obj1 Objective Status known and Objective Measure not known, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB13a
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.success_status to `failed`; Process a *Continue* navigation request	Identify Activity 4 for delivery

Test Case: OB-13b

Activity Tree Structure:



Sequencing Information:

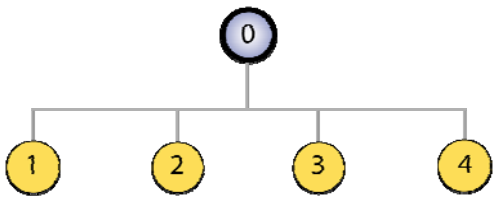
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB13b Write Satisfied Status == true Write Normalized Measure = true
3	Sequencing Rules: Pre Condition Rule: If obj1 Objective Status not known and Objective Measure known, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB13b
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.4`; Process a *Continue* navigation request	Identify Activity 4 for delivery

Test Case: OB-13c

Activity Tree Structure:



Sequencing Information:

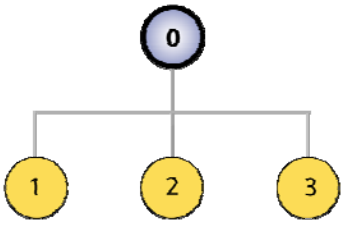
Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Default
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.25 Map Info: Target Objective ID == gObj-OB13c Write Normalized Measure = true Write Satisfied Status == true
3	Sequencing Rules: Pre Condition Rule: If obj1 Objective Status known and Objective Measure known, then skip Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB13c
4	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of `obj1`; Set that objective's cmi.objectives.n.score.scaled to `0.4`; Process a *Continue* navigation request	Identify Activity 4 for delivery

Test Case: OB-14a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-OB14a-2 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-OB14a-3 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj4 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.4 Map Info: Target Objective ID == gObj-OB14a-4 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB14a-1

	Write Normalized Measure == true Write Satisfied Status == true
2	Objectives: Primary Objective: *empty* Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-OB14a-3 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj5 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.3 Map Info: Target Objective ID == gObj-OB14a-3 Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB14a-1 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj4 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.15 Map Info: Target Objective ID == gObj-OB14a-4 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-OB14a-2 Write Normalized Measure == true Write Satisfied Status == true
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj4 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-OB14a-4 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB14a-1 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj2

	Map Info: Target Objective ID == gObj-OB14a-2 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-OB14a-3 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj5 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.01 Map Info: Target Objective ID == gObj-OB14a-3 Delivery Controls: Tracked == false

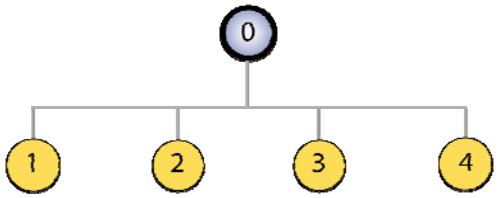
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.score.scaled to `0.5`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 1's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 1's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.score.scaled to `0.1`; Find the index of Activity 1's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to `0.2`; Set Activity 1's cmi.session_time to `PT1H0.15S`; Set Activity 1's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Find the index of Activity 2's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.score.scaled to `0.99`; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 2's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 2's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.score.scaled to `0.25`; Find the index of Activity 2's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to `0.3`; Set Activity 2's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Find the index of Activity 3's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.score.scaled to `-0.2`; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 3's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 3's cmi.objectives.n with ID of obj3;	Identify Activity 2 for delivery

	Set that objective's cmi.objectives.n.score.scaled to -0.8; Find the index of Activity 3's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to 0.65; Process a *Previous* navigation request	
5.	Find the index of Activity 2's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.score.scaled to 0.4; Find the index of Activity 2's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.score.scaled to 0.29; Find the index of Activity 2's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to 0.39; Process a *Previous* navigation request	Identify Activity 1 for delivery
6.	Find the index of Activity 1's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to passed; Find the index of Activity 1's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to 0.6; Set Activity 1's cmi.session_time to PT23H0.4S; Set Activity 1's cmi.exit to suspend; Process a *Continue* navigation request	Identify Activity 2 for delivery
7.	Find the index of Activity 2's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to failed; Find the index of Activity 2's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to 0.1; Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: OB-14b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Objectives: Primary Objective: *empty* Objective: Objective ID == obj4 Objective: Objective ID == obj1 Objective: Objective ID == obj3 Objective: Objective ID == obj2 Objective Satisfied by Measure == true Minimum Normalized Measure == 0.5 Map Info: Target Objective ID == gObj-OB14b-1 Write Normalized Measure == true Write Satisfied Status == true
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB14b-2 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj3 Objective: Objective ID == obj2 objective: Objective ID == obj1 Objective:

	Objective ID == obj5 Map Info: Target Objective ID == gObj-OB14b-3 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj4
3	Objectives: Primary Objective: *empty* Objective: Objective ID == obj5 Objective: Objective ID == obj4 Map Info: Target Objective ID == gObj-OB14b-1 Write Normalized Measure == true Write Satisfied Status == true Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-OB14b-2 Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-OB14b-3 Objective: Objective ID == obj2 Delivery Controls: Tracked == false
4	Objectives: Primary Objectives: *empty* Objective: Objective ID ==obj1 Map Info: Target Objective ID == gObj-OB14b-3 Write Normalized Measure == true Write Satisfied Status == true Read Normalized Measure == false Read Satisfied Status == false Map Info: Target Objective ID == gObj-OB14b-1 Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-OB14b-2

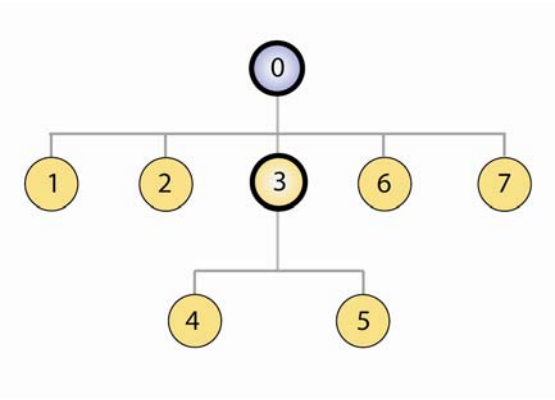
Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Find the index of Activity 1's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.success_status to `failed`; Find the index of Activity 1's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.score.scaled to `0.7`; Find the index of Activity 1's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.success_status to `failed`; Set that objective's cmi.objectives.n.score.scaled to `0.4`; Find the index of Activity 1's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to `0.25`; Set Activity 1's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 2 for delivery
3.	Set that objective's cmi.score.scaled to `0.7`; Find the index of Activity 2's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.score.scaled to `0.9`; Find the index of Activity 2's cmi.objectives.n with ID of obj2; Set that objective's cmi.objectives.n.success_status to `passed`; Find the index of Activity 2's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.success_status to `failed`; Set that objective's cmi.objectives.n.score.scaled to `0.75`; Find the index of Activity 2's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.score.scaled to `0.1`; Find the index of Activity 2's cmi.objectives.n with ID of obj5; Set that objective's cmi.objectives.n.success_status to `failed`; Set that objective's cmi.objectives.n.score.scaled to `0.2`; Set Activity 2's cmi.exit to `suspend`; Process a *Continue* navigation request	Identify Activity 3 for delivery
4.	Find the index of Activity 3's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.success_status to `passed`; Set that objective's cmi.objectives.n.score.scaled to `0.9`; Find the index of Activity 3's cmi.objectives.n with ID of obj2;	Identify Activity 4 for delivery

	Set that objective's cmi.objectives.n.score.scaled to `0.99`; Find the index of Activity 3's cmi.objectives.n with ID of obj3; Set that objective's cmi.objectives.n.success_status to `failed`; Set that objective's cmi.objectives.n.score.scaled to `0.2`; Find the index of Activity 3's cmi.objectives.n with ID of obj4; Set that objective's cmi.objectives.n.success_status to `passed`; Set that objective's cmi.objectives.n.score.scaled to `0.98`; Find the index of Activity 3's cmi.objectives.n with ID of obj5; Set that objective's cmi.objectives.n.success_status to `failed`; Set Activity 3's cmi.exit to `suspend`; Process a *Continue* navigation request	
5.	Find the index of Activity 4's cmi.objectives.n with ID of obj1; Set that objective's cmi.objectives.n.success_status to `passed`; Set that objective's cmi.objectives.n.score.scaled to `-0.5`; Process a *Previous* navigation request	Identify Activity 3 for delivery
6.	Process a *Previous* navigation request	Identify Activity 2 for delivery
7.	Process a *Previous* navigation request	Identify Activity 1 for delivery

Test Case: OB-15

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false
1	Sequencing Rules: Pre Condition Rule: If Attempt Progress not known, then skip
2	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB15 Write Satisfied Status == true
3	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If completed, then skip Rollup Rules: Completed if all Satisfied
4	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB15
5	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB15

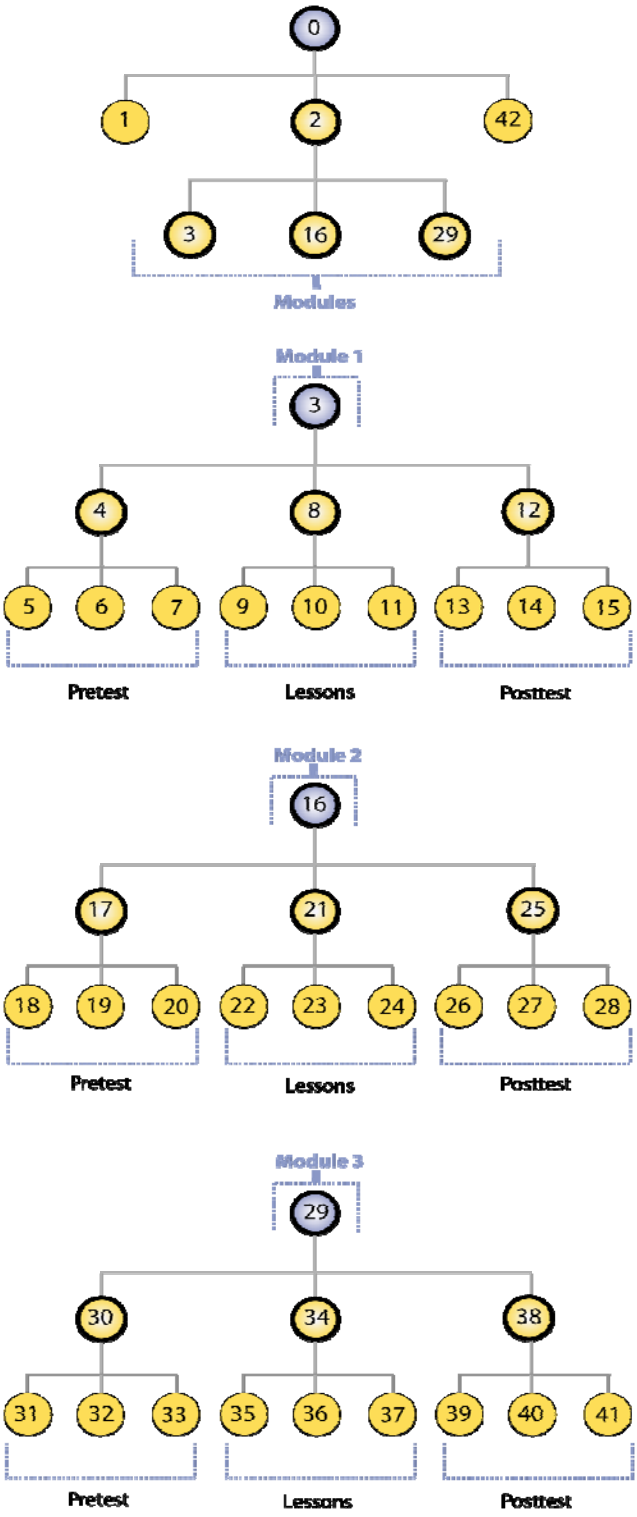
6	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-OB15 Sequencing Rules: Pre Condition Rule: If Attempt Progress not known, then skip Rollup Rules: Completed if all Satisfied
7	Default

Test Script:

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 2 for delivery
2.	Set Activity 2's cmi.success_status to `passed`; Process a *Continue* navigation request	Identify Activity 7 for delivery

Test Case: T-01a

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Rollup Rules: If any not satisfied, then not satisfied
1	Rollup Rules: Rollup Objective Satisfied == false
2	Control Model: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If always, then continue
3	***Include Module Collection***
4	***Include Pretest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-1 Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true
8	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01a-1
12	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-1 Write Satisfied Status == true Write Normalized Measure == true
16	***Include Module Collection***
17	***Include Pretest Collection*** Objectives: Primary Objective:

Activity	Sequencing Information
	Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-2 Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true
21	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01a-2
25	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-2 Write Satisfied Status == true Write Normalized Measure == true
29	***Include Module Collection***
30	***Include Pretest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-3 Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true
34	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01a-3
38	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01a-3 Write Satisfied Status == true Write Normalized Measure == true
42	Limit Conditions: Attempt limit == 1

Activity	Sequencing Information
	Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-T01a-1 Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-T01a-2 Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-T01a-3

Sequencing Collection	Sequencing Information
Module	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit
Pretest	Control Mode: Flow == true Choice == false Forward Only == true Sequencing Rules: Pre Condition Rule: If satisfied, then skip Limit Conditions: Attempt limit == 1 Rollup Rules: Completed if all attempted Rollup Considerations: Required for Completion if attempted
Lessons	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If satisfied, then skip Rollup Rules: Rollup Objective Satisfied == false Rollup Progress Completion == false
Posttest	Control Mode: Flow == true Choice == false Forward Only == true Sequencing Rules: Pre Condition Rule:

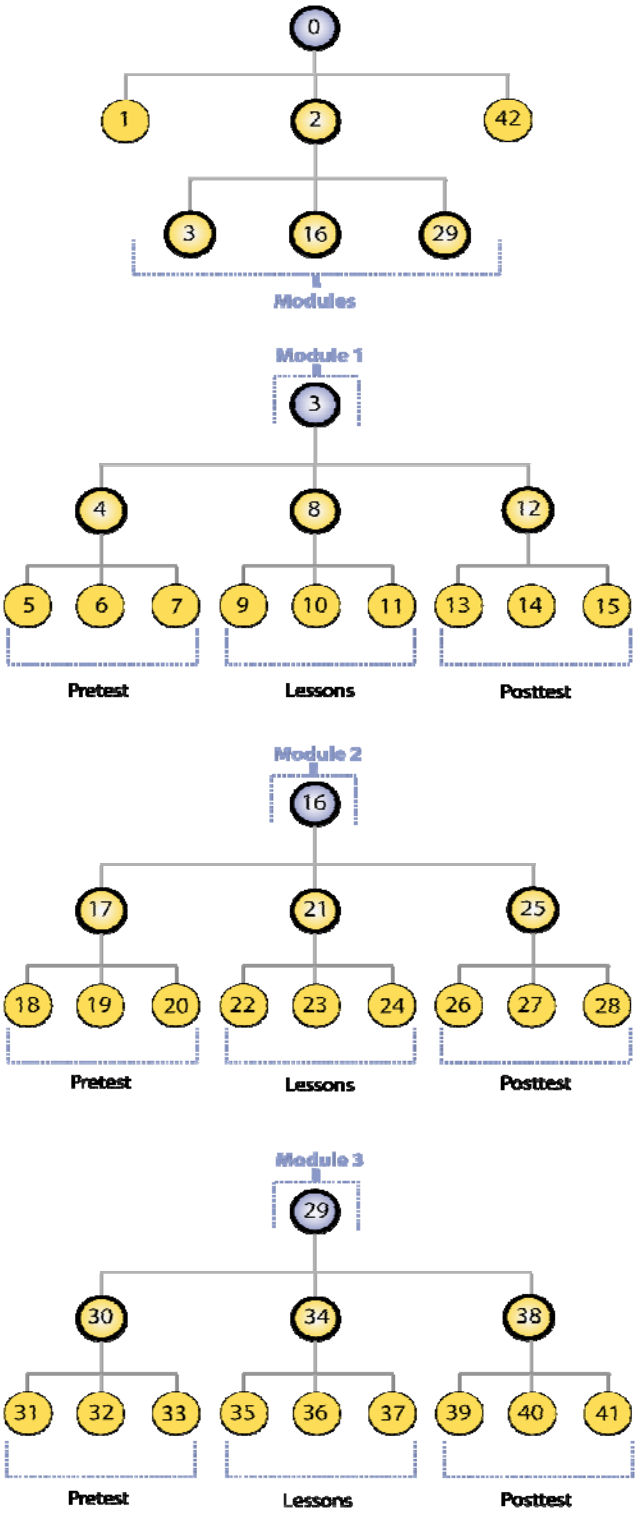
Sequencing Collection	Sequencing Information
	If satisfied, then skip Limit Conditions: Attempt limit == 1 Rollup Rules: Completed if all attempted Rollup Considerations: Required for Completion if not skipped

Test Script: T-01a

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 5 for delivery
3.	Set Activity 5's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 6 for delivery
4.	Set Activity 6's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 7 for delivery
5.	Set Activity 7's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 18 for delivery
6.	Set Activity 18's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 19 for delivery
7.	Set Activity 19's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 20 for delivery
8.	Set Activity 20's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 31 for delivery
9.	Set Activity 31's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 32 for delivery
10.	Set Activity 32's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 33 for delivery
11.	Set Activity 33's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 42 for delivery

Test Case: T-01b

Activity Tree Structure:



Sequencing Information:

Activity	Sequencing Information
0	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Rollup Rules: If any not satisfied, then not satisfied
1	Rollup Rules: Rollup Objective Satisfied == false
2	Control Model: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit Post Condition Rule: If always, then continue
3	***Include Module Collection***
4	***Include Pretest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-1 Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true
8	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01b-1
9	Objectives: Primary Objective: *empty* Objective: Objective ID == obj1 Delivery Controls: Tracked == false
10	Objectives: Primary Objective: *empty* Objective: Objective ID == obj2 Objective: Objective ID == obj3 Objective: Objective ID == obj4

Activity	Sequencing Information
	Objective: Objective ID == obj5 Objective: Objective Id == obj6 Delivery Controls: Tracked == false
12	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-1 Write Satisfied Status == true Write Normalized Measure == true
16	***Include Module Collection***
17	***Include Pretest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-2 Read Normalized Measure == false Write Satisfied Status == true Write Normalized Measure == true
21	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01b-2
25	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-2 Write Satisfied Status == true Write Normalized Measure == true
29	***Include Module Collection***
30	***Include Pretest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-3 Read Normalized Measure == false

Activity	Sequencing Information
	Write Satisfied Status == true Write Normalized Measure == true
34	***Include Lessons Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Map Info: Target Objective ID == gObj-T01b-3
38	***Include Posttest Collection*** Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective Satisfied by Measure == true Minimum Normalized Measure == 0.6 Map Info: Target Objective ID == gObj-T01b-3 Write Satisfied Status == true Write Normalized Measure == true
42	Limit Conditions: Attempt limit == 1 Objectives: Primary Objective: Objective ID == PRIMARYOBJ Objective: Objective ID == obj1 Map Info: Target Objective ID == gObj-T01b-1 Objective: Objective ID == obj2 Map Info: Target Objective ID == gObj-T01b-2 Objective: Objective ID == obj3 Map Info: Target Objective ID == gObj-T01b-3

Sequencing Collection	Sequencing Information
Module	Control Mode: Flow == true Choice == false Sequencing Rules: Exit Rule: If completed, then exit
Pretest	Control Mode: Flow == true Choice == false Forward Only == true Sequencing Rules: Pre Condition Rule: If satisfied, then skip Limit Conditions:

Sequencing Collection	Sequencing Information
	Attempt limit == 1 Rollup Rules: Completed if all attempted Rollup Considerations: Required for Completion if attempted
Lessons	Control Mode: Flow == true Choice == false Sequencing Rules: Pre Condition Rule: If satisfied, then skip Rollup Rules: Rollup Objective Satisfied == false Rollup Progress Completion == false
Posttest	Control Mode: Flow == true Choice == false Forward Only == true Sequencing Rules: Pre Condition Rule: If satisfied, then skip Limit Conditions: Attempt limit == 1 Rollup Rules: Completed if all attempted Rollup Considerations: Required for Completion if not skipped

Test Script: T-01b

Step	Action	Expected Result
1.	Process a *Start* navigation request	Identify Activity 1 for delivery
2.	Process a *Continue* navigation request	Identify Activity 5 for delivery
3.	Set Activity 5's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 6 for delivery
4.	Set Activity 6's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 7 for delivery
5.	Set Activity 7's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 9 for delivery
6.	Process a *Continue* navigation request	Identify Activity 10 for delivery
7.	Process a *Continue* navigation request	Identify Activity 11 for delivery
8.	Process a *Continue* navigation request	Identify Activity 13 for delivery
9.	Set Activity 13's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 14 for delivery
10.	Set Activity 14's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 18 for delivery
11.	Set Activity 18's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 19 for delivery
12.	Set Activity 19's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 20 for delivery
13.	Set Activity 20's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 31 for delivery
14.	Set Activity 31's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 32 for delivery
15.	Set Activity 32's cmi.score.scaled to `1.0`; Process a *Continue* navigation request	Identify Activity 33 for delivery
16.	Set Activity 33's cmi.score.scaled to `0.0`; Process a *Continue* navigation request	Identify Activity 42 for delivery

This page intentionally left blank.

APPENDIX B
ISO-639 and IANA Language Codes

This page intentionally left blank.

ISO-639 and IANA Language Codes

ISO 639-1 Codes

Obtained from the ISO 639-1 Registration Authority – Infoterm (http://linux.infoterm.org/infoterm-e/i-infoterm.htm?raiso639-1_start.htm~Mitte) on September-30-2004.

Abkhazian	ab
Afar	aa
Afrikaans	af
Akan	ak
Albanian	sq
Amharic	am
Arabic	ar
Aragonese	an
Armenian	hy
Assamese	as
Avaric	av
Avestan	ae
Aymara	ay
Azerbaijani	az
Bambara	bm
Bashkir	ba
Basque	eu
Belarusian	be
Bengali	bn
Bihari	bh
Bislama	bi
Bokmål, Norwegian; Norwegian Bokmål	nb
Bosnian	bs
Breton	br
Bulgarian	bg
Burmese	my
Castilian; Spanish	es
Catalan; Valencian	ca
Chamorro	ch
Chechen	ce
Chewa; Chichewa; Nyanja	ny
Chichewa; Chewa; Nyanja	ny
Chinese	zh
Chuang; Zhuang	za
Church Slavic; Slavonic; Church Slavonic; Old Bulgarian; Old Church Slavonic	cu
Church Slavonic; Church Slavic; Old Slavonic; Old Bulgarian; Old Church Slavonic	cu
Chuvash	cv
Cornish	kw

Corsican	co
Cree	cr
Croatian	hr
Czech	cs
Danish	da
Divehi	dv
Dutch; Flemish	nl
Dzongkha	dz
English	en
Esperanto	eo
Estonian	et
Ewe	ee
Faroese	fo
Fijian	fj
Finnish	fi
Flemish; Dutch	nl
French	fr
Frisian	fy
Fulah	ff
Gaelic; Scottish Gaelic	gd
Gallegan	gl
Ganda	lg
Georgian	ka
German	de
Gikuyu; Kikuyu	ki
Greek, Modern (1453-)	el
Greenlandic; Kalaallisut	kl
Guarani	gn
Gujarati	gu
Haitian; Haitian Creole	ht
Haitian Creole; Haitian	ht
Hausa	ha
Hebrew	he
Herero	hz
Hindi	hi
Hiri Motu	ho
Hungarian	hu
Icelandic	is
Ido	io
Igbo	ig
Indonesian	id
Interlingua (International Auxiliary Language Association)	ia
Interlingue	ie
Inuktitut	iu
Inupiaq	ik
Irish	ga
Italian	it
Japanese	ja

Javanese	jv
Kalaallisut; Greenlandic	kl
Kannada	kn
Kanuri	kr
Kashmiri	ks
Kazakh	kk
Khmer	km
Kikuyu; Gikuyu	ki
Kinyarwanda	rw
Kirghiz	ky
Komi	kv
Kongo	kg
Korean	ko
Kuanyama; Kwanyama	kj
Kurdish	ku
Kwanyama, Kuanyama	kj
Lao	lo
Latin	la
Latvian	lv
Letzeburgesch; Luxembourgish	lb
Limburgan; Limburger; Limburgish	li
Limburger; Limburgan; Limburgish;	li
Limburgish; Limburger; Limburgan	li
Lingala	ln
Lithuanian	lt
Luba-Katanga	lu
Luxembourgish; Letzeburgesch	lb
Macedonian	mk
Malagasy	mg
Malay	ms
Malayalam	ml
Maltese	mt
Manx	gv
Maori	mi
Marathi	mr
Marshallese	mh
Moldavian	mo
Mongolian	mn
Nauru	na
Navaho, Navajo	nv
Navajo; Navaho	nv
Ndebele, North	nd
Ndebele, South	nr
Ndonga	ng
Nepali	ne
Northern Sami	se
North Ndebele	nd
Norwegian	no

Norwegian Bokmål; Bokmål, Norwegian	nb
Norwegian Nynorsk; Nynorsk, Norwegian	nn
Nyanja; Chichewa; Chewa	ny
Nynorsk, Norwegian; Norwegian Nynorsk	nn
Occitan (post 1500); Provençal	oc
Ojibwa	oj
Old Bulgarian; Old Slavonic; Church Slavonic; Church Slavic; Old Church Slavonic	cu
Old Church Slavonic; Old Slavonic; Church Slavonic; Old Bulgarian; Church Slavic	cu
Old Slavonic; Church Slavonic; Old Bulgarian; Church Slavic; Old Church Slavonic	cu
Oriya	or
Oromo	om
Ossetian; Ossetic	os
Ossetic; Ossetian	os
Pali	pi
Panjabi; Punjabi	pa
Persian	fa
Polish	pl
Portuguese	pt
Provençal; Occitan (post 1500)	oc
Punjabi; Panjabi	pa
Pushto	ps
Quechua	qu
Raeto-Romance	rm
Romanian	ro
Rundi	rn
Russian	ru
Samoan	sm
Sango	sg
Sanskrit	sa
Sardinian	sc
Scottish Gaelic; Gaelic	gd
Serbian	sr
Shona	sn
Sichuan Yi	ii
Sindhi	sd
Sinhalese	si
Slovak	sk
Slovenian	sl
Somali	so
Sotho, Southern	st
South Ndebele	nr
Spanish; Castilian	es
Sundanese	su
Swahili	sw
Swati	ss
Swedish	sv
Tagalog	tl
Tahitian	ty

Tajik	tg
Tamil	ta
Tatar	tt
Telugu	te
Thai	th
Tibetan	bo
Tigrinya	ti
Tonga (Tonga Islands)	to
Tsonga	ts
Tswana	tn
Turkish	tr
Turkmen	tk
Twi	tw
Uighur	ug
Ukrainian	uk
Urdu	ur
Uzbek	uz
Valencian; Catalan	ca
Venda	ve
Vietnamese	vi
Volapük	vo
Walloon	wa
Welsh	cy
Wolof	wo
Xhosa	xh
Yiddish	yi
Yoruba	yo
Zhuang; Chuang	za
Zulu	zu

ISO 639-2 Codes

Obtained from the ISO 639-2 Registration Authority – Library of Congress (http://www.loc.gov/standards/iso639-2/langhome.html) on September-30-2004.

Abkhazian	abk
Achinese	ace
Acoli	ach
Adangme	ada
Adygei	ady
Adyghe	ady
Afar	aar
Afrihili	afh
Afrikaans	afr
Afro-Asiatic (Other)	afa
Akan	aka
Akkadian	akk
Albanian	alb/sqi
Aleut	ale
Algonquian languages	alg
Altaic (Other)	tut
Amharic	amh
Apache languages	apa
Arabic	ara
Aragonese	arg
Aramaic	arc
Arapaho	arp
Araucanian	arn
Arawak	arw
Armenian	arm/hye
Artificial (Other)	art
Assamese	asm
Asturian	ast
Athapascan languages	ath
Australian languages	aus
Austronesian (Other)	map
Avaric	ava
Avestan	ave
Awadhi	awa
Aymara	aym
Azerbaijani	aze
Bable	ast
Balinese	ban
Baltic (Other)	bat
Baluchi	bal
Bambara	bam
Bamileke languages	bai

Banda	bad
Bantu (Other)	bnt
Basa	bas
Bashkir	bak
Basque	baq/eus
Batak (Indonesia)	btk
Beja	bej
Belarusian	bel
Bemba	bem
Bengali	ben
Berber (Other)	ber
Bhojpuri	bho
Bihari	bih
Bikol	bik
Bilin	byn
Bini	bin
Bislama	bis
Blin	byn
Bokmål, Norwegian	nob
Bosnian	bos
Braj	bra
Breton	bre
Buginese	bug
Bulgarian	bul
Buriat	bua
Burmese	bur/mya
Caddo	cad
Carib	car
Castilian	spa
Catalan	cat
Caucasian (Other)	cau
Cebuano	ceb
Celtic (Other)	cel
Central American Indian (Other)	cai
Chagatai	chg
Chamic languages	cmc
Chamorro	cha
Chechen	che
Cherokee	chr
Chewa	nya
Cheyenne	chy
Chibcha	chb
Chichewa	nya
Chinese	chi/zho
Chinook jargon	chn
Chipewyan	chp
Choctaw	cho
Chuang	zha

Church Slavic	chu
Church Slavonic	chu
Chuukese	chk
Chuvash	chv
Classical Nepal Bhasa	nwc
Classical Newari	nwc
Coptic	cop
Cornish	cor
Corsican	cos
Cree	cre
Creek	mus
Creoles and pidgins(Other)	crp
Creoles and pidgins, English-based (Other)	cpe
Creoles and pidgins, French-based (Other)	cpf
Creoles and pidgins, Portuguese-based (Other)	cpp
Crimean Tatar	crh
Crimean Turkish	crh
Croatian	scr/hrv
Cushitic (Other)	cus
Czech	cze/ces
Dakota	dak
Danish	dan
Dargwa	dar
Dayak	day
Delaware	del
Dinka	din
Divehi	div
Dogri	doi
Dogrib	dgr
Dravidian (Other)	dra
Duala	dua
Dutch	dut/nld
Dutch, Middle (ca. 1050-1350)	dum
Dyula	dyu
Dzongkha	dzo
Efik	efi
Egyptian (Ancient)	egy
Ekajuk	eka
Elamite	elx
English	eng
English, Middle (1100-1500)	enm
English, Old (ca.450-1100)	ang
Erzya	myv
Esperanto	epo
Estonian	est
Ewe	ewe
Ewondo	ewo
Fang	fan

Fanti	fat
Faroese	fao
Fijian	fij
Filipino	fil
Finnish	fin
Finno-Ugrian (Other)	fiu
Fon	fon
French	fre/fra
French, Middle (ca.1400-1600)	frm
French, Old (842-ca.1400)	fro
Frisian	fry
Friulian	fur
Fulah	ful
Ga	gaa
Gaelic	gla
Gallegan	glg
Ganda	lug
Gayo	gay
Gbaya	gba
Geez	gez
Georgian	geo/kat
German	ger/deu
German, Low	nds
German, Middle High (ca.1050-1500)	gmh
German, Old High (ca.750-1050)	goh
Germanic (Other)	gem
Gikuyu	kik
Gilbertese	gil
Gondi	gon
Gorontalo	gor
Gothic	got
Grebo	grb
Greek, Ancient (to 1453)	grc
Greek, Modern (1453-)	gre/ell
Guarani	grn
Gujarati	guj
Gwich´in	gwi
Haida	hai
Haitian	hat
Haitian Creole	hat
Hausa	hau
Hawaiian	haw
Hebrew	heb
Herero	her
Hiligaynon	hil
Himachali	him
Hindi	hin
Hiri Motu	hmo

Hittite	hit
Hmong	hmn
Hungarian	hun
Hupa	hup
Iban	iba
Icelandic	ice/isl
Ido	ido
Igbo	ibo
Ijo	ijo
Iloko	ilo
Inari Sami	smn
Indic (Other)	inc
Indo-European (Other)	ine
Indonesian	ind
Ingush	inh
Interlingua (International Auxiliary Language Association)	ina
Interlingue	ile
Inuktitut	iku
Inupiaq	ipk
Iranian (Other)	ira
Irish	gle
Irish, Middle (900-1200)	mga
Irish, Old (to 900)	sga
Iroquoian languages	iro
Italian	ita
Japanese	jpn
Javanese	jav
Judeo-Arabic	jrb
Judeo-Persian	jpr
Kabardian	kbd
Kabyle	kab
Kachin	kac
Kalaallisut	kal
Kalmyk	xal
Kamba	kam
Kannada	kan
Kanuri	kau
Karachay-Balkar	krc
Kara-Kalpak	kaa
Karen	kar
Kashmiri	kas
Kashubian	csb
Kawi	kaw
Kazakh	kaz
Khasi	kha
Khmer	khm
Khoisan (Other)	khi
Khotanese	kho

Kikuyu	kik
Kimbundu	kmb
Kinyarwanda	kin
Kirghiz	kir
Klingon	tlh
Komi	kom
Kongo	kon
Konkani	kok
Korean	kor
Kosraean	kos
Kpelle	kpe
Kru	kro
Kuanyama	kua
Kumyk	kum
Kurdish	kur
Kurukh	kru
Kutenai	kut
Kwanyama	kua
Ladino	lad
Lahnda	lah
Lamba	lam
Lao	lao
Latin	lat
Latvian	lav
Letzeburgesch	ltz
Lezghian	lez
Limburgan	lim
Limburger	lim
Limburgish	lim
Lingala	lin
Lithuanian	lit
Lojban	jbo
Low German	nds
Low Saxon	nds
Lower Sorbian	dsb
Lozi	loz
Luba-Katanga	lub
Luba-Lulua	lua
Luiseno	lui
Lule Sami	smj
Lunda	lun
Luo (Kenya and Tanzania)	luo
Luxembourgish	ltz
Lushai	lus
Macedonian	mac/mkd
Madurese	mad
Magahi	mag
Maithili	mai

Makasar	mak
Malagasy	mlg
Malay	may/msa
Malayalam	mal
Maltese	mlt
Manchu	mnc
Mandar	mdr
Mandingo	man
Manipuri	mni
Manobo languages	mno
Manx	glv
Maori	mao/mri
Marathi	mar
Mari	chm
Marshallese	mah
Marwari	mwr
Masai	mas
Mayan languages	myn
Mende	men
Micmac	mic
Minangkabau	min
Mirandese	mwl
Miscellaneous languages	mis
Mohawk	moh
Moksha	mdf
Moldavian	mol
Mon-Khmer (Other)	mkh
Mongo	lol
Mongolian	mon
Mossi	mos
Multiple languages	mul
Munda languages	mun
Nahuatl	nah
Nauru	nau
Navaho	nav
Navajo	nav
Ndebele, North	nde
Ndebele, South	nbl
Ndonga	ndo
Neapolitan	nap
Nepal Bhasa	new
Nepali	nep
Newari	new
Nias	nia
Niger-Kordofanian (Other)	nic
Nilo-Saharan (Other)	ssa
Niuean	niu
Nogai	nog

Norse, Old	non
North American Indian (Other)	nai
Northern Sami	sme
North Ndebele	nde
Norwegian	nor
Norwegian Bokmål	nob
Norwegian Nynorsk	nno
Nubian languages	nub
Nyamwezi	nym
Nyanja	nya
Nyankole	nyn
Nynorsk, Norwegian	nno
Nyoro	nyo
Nzima	nzi
Occitan (post 1500)	oci
Ojibwa	oji
Old Bulgarian	chu
Old Church Slavonic	chu
Old Newari	nwc
Old Slavonic	chu
Oriya	ori
Oromo	orm
Osage	osa
Ossetian	oss
Ossetic	oss
Otomian languages	oto
Pahlavi	pal
Palauan	pau
Pali	pli
Pampanga	pam
Pangasinan	pag
Panjabi	pan
Papiamento	pap
Papuan (Other)	paa
Persian	per/fas
Persian, Old (ca.600-400)	peo
Philippine (Other)	phi
Phoenician	phn
Pilipino	fil
Pohnpeian	pon
Polish	pol
Portuguese	por
Prakrit languages	pra
Provençal	oci
Provençal, Old (to 1500)	pro
Punjabi	pan
Pushto	pus
Quechua	que

Raeto-Romance	roh
Rajasthani	raj
Rapanui	rap
Rarotongan	rar
Reserved for local user	qaa-qtz
Romance (Other)	roa
Romanian	rum/ron
Romany	rom
Rundi	run
Russian	rus
Salishan languages	sal
Samaritan Aramaic	sam
Sami languages (Other)	smi
Samoan	smo
Sandawe	sad
Sango	sag
Sanskrit	san
Santali	sat
Sardinian	srd
Sasak	sas
Saxon, Low	nds
Scots	sco
Scottish Gaelic	gla
Selkup	sel
Semitic (Other)	sem
Serbian	scc/srp
Serer	srr
Shan	shn
Shona	sna
Sichuan Yi	iii
Sicilian	scn
Sidamo	sid
Sign languages	sgn
Siksika	bla
Sindhi	snd
Sinhalese	sin
Sino-Tibetan (Other)	sit
Siouan languages	sio
Skolt Sami	sms
Slave (Athapascan)	den
Slavic (Other)	sla
Slovak	slo/slk
Slovenian	slv
Sogdian	sog
Somali	som
Songhai	son
Soninke	snk
Sorbian languages	wen

Sotho, Northern	nso
Sotho, Southern	sot
South American Indian (Other)	sai
Southern Sami	sma
South Ndebele	nbl
Spanish	spa
Sukuma	suk
Sumerian	sux
Sundanese	sun
Susu	sus
Swahili	swa
Swati	ssw
Swedish	swe
Syriac	syr
Tagalog	tgl
Tahitian	tah
Tai (Other)	tai
Tajik	tgk
Tamashek	tmh
Tamil	tam
Tatar	tat
Telugu	tel
Tereno	ter
Tetum	tet
Thai	tha
Tibetan	tib/bod
Tigre	tig
Tigrinya	tir
Timne	tem
Tiv	tiv
Tlhlngan-Hol	tlh
Tlingit	tli
Tok Pisin	tpi
Tokelau	tkl
Tonga (Nyasa)	tog
Tonga (Tonga Islands)	ton
Tsimshian	tsi
Tsonga	tso
Tswana	tsn
Tumbuka	tum
Tupi languages	tup
Turkish	tur
Turkish, Ottoman (1500-1928)	ota
Turkmen	tuk
Tuvalu	tvl
Tuvinian	tyv
Twi	twi
Udmurt	udm

Ugaritic	uga
Uighur	uig
Ukrainian	ukr
Umbundu	umb
Undetermined	und
Upper Sorbian	hsb
Urdu	urd
Uzbek	uzb
Vai	vai
Valencian	cat
Venda	ven
Vietnamese	vie
Volapük	vol
Votic	vot
Wakashan languages	wak
Walamo	wal
Walloon	wln
Waray	war
Washo	was
Welsh	wel/cym
Wolof	wol
Xhosa	xho
Yakut	sah
Yao	yao
Yapese	yap
Yiddish	yid
Yoruba	yor
Yupik languages	ypk
Zande	znd
Zapotec	zap
Zenaga	zen
Zhuang	zha
Zulu	zul
Zuni	zun

IANA Registered Language Codes

Obtained from IANA (http://www.iana.org/assignments/lang-tag-apps.htm) on September-30-2004.

art-lojban
az-Arab
az-Cyrl
az-Latn
cel-gaulish
de-1901
de-1996
de-AT-1901
de-AT-1996
de-CH-1901
de-CH-1996
de-DE-1901
de-DE-1996
en-boont
en-GB-oed
en-scouse
i-ami
i-bnn
i-default
i-enochian
i-hak
i-klingon
i-lux
i-mingo
i-navajo
i-pwn
i-tao
i-tay
i-tsu
no-bok
no-nyn
sgn-BE-fr
sgn-BE-nl
sgn-BR
sgn-CH-de
sgn-CO
sgn-DE
sgn-DK
sgn-ES
sgn-FR
sgn-GB
sgn-GR
sgn-IE
sgn-IT
sgn-JP
sgn-MX
sgn-NL

sgn-NO
sgn-PT
sgn-SE
sgn-US
sgn-ZA
sl-rozaj
sr-Cyrl
sr-Latn
uz-Cyrl
uz-Latn
yi-Latn
zh-gan
zh-guoyu
zh-hakka
zh-Hans
zh-Hant
zh-min
zh-min-nan
zh-wuu
zh-xiang

APPENDIX C
Acronym Listing

This page intentionally left blank.

Acronym Listing

ADL	Advanced Distributed Learning
API	Application Program Interface
CAM	Content Aggregation Model
CP	Content Packaging
CR	Conformance Requirements
DoD	Department of Defense
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
LMS	Learning Management System
MD	Metadata
MIME	Multipurpose Internet Mail Extensions
PIF	Package Interchange Format
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
URN	Universal Resource Name
XSD	XML Schema Definition
XML	Extensible Markup Language

This page intentionally left blank.

APPENDIX D
References

This page intentionally left blank.

References

1. Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition
   Includes:
   The SCORM 2004 3rd Edition Overview
   The SCORM 2004 3rd Edition Content Aggregation Model (CAM) Version 1.0
   The SCORM 2004 3rd Edition Run-Time Environment (RTE) Version 1.0
   The SCORM 2004 3rd Edition Sequencing and Navigation (SN) Version 1.0
   Available at http://www.adlnet.gov/

This page intentionally left blank.

APPENDIX E
Document Revision History

This page intentionally left blank.

Document Revision History

Version	Release Date	Description of Change
1.0	31-Mar-2004	Initial release.
1.1	22-Jul-2004	Updates to reflect the following: • Various grammar and style changes. • Updated the Conformance Categories in the Conformance Requirements Matrix to reflect version changes to the books described in SCORM 2004 2nd Edition. • Updated requirements for GetErrorString() and GetDiagnostic() to indicate that the value returned from these API methods shall have a maximum length of 255 characters. • Updated REQ_57.5 and related requirements to change data model element name from cmi.comments_from_learner.n.date_time to cmi.comments_from_learner.n.timestamp. • Updated REQ_58.5 and related requirements to change data model element name from cmi.comments_from_lms.n.date_time to cmi.comments_from_lms.n.timestamp. • Updated requirements impacted by the change from time (second,10,2) to time (second,10,0). These requirements are: REQ_57.5.2, REQ_58.5.2, REQ_64.6.2. • Updated REQ_62.3.2 to include more information on describing a suspended learner session. • Added requirements that explicitly spell out requirements dealing with LMS handling of correct responses per interaction type. • Added REQ_68.5.3 to describe the requirement of initializing the cmi.learner_preference.audio_level to its default of 1, if no mechanism is in place to initialize this value. • Updated requirements to clarify the initialization of cmi.objectives.n.xxx data based on objectives defined in a manifest. • Added REQ_72.8 to describe requirements for processing and handling of the cmi.objectives.n.progress_measure. • Update REQ_79.3 to define the requirement of initializing the cmi.time_limit_action to the default of “continue,no message” if the <adlcp:timeLimitAction> is not defined in a manifest. • Added REQ_83.2 that details the SPM for the value represented by the <language_type>, SPM of 250 characters • Added REQ_84.3 that details the SPM for the value represented by a long_identifier_type, SPM of 4000 characters • Added REQ_84.3 that details the SPM for the value represented by a short_identifier_type, SPM of 250 characters

Version	Release Date	Description of Change
		• Updated REQ_88 to change time (second,10,2) to time (second,10,0). • Updated REQ_88.1 to add more guidance and clarification on the requirements of the TZD portion of a time (second,10,0). • Updated REQ_91 and its child requirements to clarify and add requirements based on SCORM 2004 updates. • Added REQ_30.6.3.6.13, REQ_30.6.3.6.13.1 and REQ_30.6.3.6.13.2. These requirements describe the missing requirements for the <adlcp:completionThreshold> element • Updated REQ_31.1 to state that the <sequencingCollection> shall contain 1 or More (instead of 0 or More) <sequencing> elements. • Added REQ_31.2.1 to describe that leaf item's that reference (sub)manifests shall not have <sequencing> information. • Added REQ_31.3.3 and REQ_31.3.4 to clarify the requirements for the ID element of a <sequencing> element. • Added requirements defining that objective IDs cannot be an empty characterstring or contain all white space (REQ_31.6.1.1.2.1.2, REQ_31.10.1.2.4, REQ_31.10.1.4.1.2, REQ_31.10.2.2.3). • Added additional requirements for SCO setting of correct_response patterns for the true-false, likert and numeric interaction types. (REQ_100.9.2.3) • Updated requirements based on <mapInfo> usage. • Added missing requirements that describe the initialization of cmi.objectives.n.xxx data based on objectives defined in the manifest. • Changed requirements dealing with the choice navigation data model elements from being a URI to a <STRING>. • Updated REQ_29.5.3 to reflect the multiplicity requirements for the <schemaversion> element. • Updated REQ_41 to change DateTime to Duration • Updated REQ_30.7.3.2.2 to clarify the requirement of using the value of "webcontent" for the type attribute. • Added missing requirement REQ_30.7.3.9.1.1. This requirement was added to clarify the meaning of "local" to the content package. • General clean up and formatting changes for clarity to Appendix A. • Added Sequencing Test Cases OB-4, OB-5a, OB-5b and RU-7c to test the following sequencing behaviors: o SCORM extended rollup algorithm o Retry condition on the root of the activity tree o Utilization of shared global objective state when local objectives are "satisfied by measure"

Version	Release Date	Description of Change
1.2	09-Nov-2004	LMS Conformance Requirements updates: • Added API Implementation Conformance Requirement REQ_7.1.2 to describe required behavior for invalid SetValue() calls on "to-be" created data model elements in a collection. • Added more detailed information on what is expected in return of _children GetValue() calls. • Updated several conformance requirements for grammar errors and to make wording consistent throughout. • Added Run-Time Environment Data Model Conformance Requirements REQ_72.3.3.3.1 and REQ_72.3.3.3.2. These requirements defined in more detail the required behavior for initializing the cmi.objectives.n.succes_status data model element. • Added Run-Time Environment Data Model Conformance Requirements REQ_74.3.1 and REQ_74.3.2. These requirements defined in more detail the required behavior for initializing the cmi.scaled_passing_score data model element. • Added Run-Time Environment Data Model Conformance Requirements REQ_77.5.3 to define, in more detail, the requirement for determining cmi.success_status for the case where a cmi.scaled_passing_score is known but a cmi.scaled.score is not. • Added more information for the Run-Time Environment Data Model Data Type Conformance Requirement REQ_83.1. This requirement deals with the language_type for the data model elements. Additional information was added to describe the language codes (Appendix B was added). Also added the restriction on the length of the langcode and subcode based on the underlying specifications for langcodes and subcodes. • Removed Sequencing Test Case SX-1 • Added Sequencing Test Case OB-5c to test updated "Satisfied by Measure" behavior. • Updated sequencing information related to Sequencing Test Cases T-1a and T1-b Content Package Conformance Requirements updates: • Deleted Requirements REQ_30.7.3.6, REQ_30.7.3.6.1, REQ_29.7.3.6 and REQ_29.7.3.6.1 due to the adlcp:persistState element being deprecated. See the SCORM 2nd Edition Addendum for more information. • Updated Requirement REQ_31.14.5.2. The default value for measureSatisfactionIfActive is true, not false. SCO Conformance Requirements updates: • Move requirements REQ_21, REQ_21.1, REQ_21.2 and REQ_21.3 to REQ_26.1, REQ_26.1.1, REQ_26.1.2 and REQ_26.1.3 General updates: • Various grammar and style changes. • Added ISO and IANA Language Codes in Appendix B
1.3	15-Apr-2005	LMS Conformance Requirements Updates: • Added missing Requirements REQ_6.10, REQ_6.11, REQ_6.12.

Version	Release Date	Description of Change
		These requirements describe GetValue() API Implementation behaviors, defined in SCORM 2004, dealing with error conditions that may occur. • Added missing Requirements REQ_7.13, REQ_7.14, REQ_7.15. These requirements describe SetValue() API Implementation behaviors, defined in SCORM 2004, dealing with error conditions that may occur. • For those data model elements that contain children, added a clarification note that indicates that the order of the elements returned is not significant (REQ_57.1.3, REQ_58.1.3, REQ_64.1.3, REQ_68.1.3, REQ_72.1.3, REQ_72.4.1.3 and REQ_75.1.3 • Added REQ_59.4.4 that describes LMS behavior for mapping the cmi.completion_status value of not attempted to the tracking model described in the Sequencing and Navigation book. • Added REQ_72.3.6 that describes the LMS behavior for processing a SetValue() request for the cmi.objectives.n.id data model element where the identifier is not unique. • Updated REQ_80.4 that defines that the LMS shall initialize the cmi.total_time data model element to any value that evaluates to a interval of time of zero. Content Package Conformance Requirements Updates: • Added requirements that define that the xml:base valus shall not contain any backward slashes ("\"), shall not begin with a leading slash ("/") and shall end with a trailing slash ("/"). • Updated occurrences of the word meta-data and changed it to metadata. • Added a note indicating the that <adlcp:location> element is affected by xml:base attribute if one is provided. Requirements REQ_305.4.1.2, REQ_30.6.3.6.8.1.1.2, REQ_30.6.3.7.1.1.2, REQ_30.7.3.8.1.1.2, REQ_30.7.3.9.4.1.1.2, REQ_29.5.4.1.2, REQ_29.7.3.8.1.2.2, REQ_29.7.3.9.4.1.1.2) • Added requirements indicating that any URLs used within a Manifest shall conform to the requirements defined in IETF RFC 3986. Requirements REQ_30.5.4.1.2.1, REQ_30.6.3.6.8.1.1.2.1, REQ_30.6.3.7.1.1.2.1, REQ_30.7.3.3.1, REQ_30.7.3.8.1.1.2.1, REQ_30.7.3.9.2.1, REQ_30.7.3.9.4.1.1.2.1, REQ_29.5.4.1.2.1, REQ_29.7.3.3.1, REQ_29.7.3.8.1.2.2.1, REQ_29.7.3.9.2.1, REQ_29.7.3.9.4.1.1.2.1 • Updated REQ_33 to indicate that the <adlnav:presentation> element can appear as a child of an <item> that references an Asset resource. • Updated REQ_28.1.4 to reference the correct SCORM Content Packaging Extension XSD • Added missing requirements REQ_28.1.7, REQ_28.1.8, REQ_28.1.9 to describe the requirements for validation against other SCORM referenced XSDs. Appendix A: Sequencing Conformance Requirements Updates: • Added Test Case RU-10, RU-11 and OB-6 to test behaviors dealing with non-tracked activities.

Version	Release Date	Description of Change
		• Updated Test Case SX-2 ‘s Test Script to set Activity 4’s cmi.success_status to failed. • Updated Test Cases RU-8a and RU-8b to reflect the original test intentsion. General updates: • Various grammar and style changes.
SCORM 2004 3rd Edition: CR Version 1.0	20-Oct-2006	General Updates: • Updated the version numbers in the conformance matrix to match the version numbers of the individual SCORM books. • Changed all references to adlnet.org to adlnet.gov • Changed all references to IETF RFC 2396 to IETF RFC 3986 • Updated requirement REQ_88.1 to change the range to 1970 <= YYYY <= 2038 and added language that states the default TZD is “local time” if the TZD is not provided. • Added a requirement that if TZD is defined then hh:mm:ss.s shall be defined, although the values may all be zeros. • Updated requirement REQ_89.1 to add the requirement that Zero – padding of values shall be supported and add more language to clarify the requirements around the syntax of the time interval. • Updated REQ_91.1.7.2.2.2.3 to change the number to REQ_91.2.7.2.2.3 Learning Management System Requirements Updates: • REQ_22.1 and REQ_22.2: updates were made to the grammar of these two requirements.] • Updated the use of the word character string to characterstring • REQ_6.10, REQ_6.11, REQ_6.12, REQ_7.13, REQ_7.14: added an ADL Note to reference the SCORM Run-Time Environment for more specific situations. • REQ_10.3, REQ_10.4: removed the parentheticals found after the word parameter. • REQ_57.4.3: Changed the requirement to indicate that this is a GetValue() requirement not a SetValue() requirement. Because of this, change the expected error code from 351 to 301. • REQ_59.5, REQ_59.5.1 and REQ_59.5.2: based on the SCORM Technical Working Group meeting, updated these requirements to clearly define the LMS behavior for evaluation of the complete status data model element. • REQ_59.5.3 and REQ_59.5.4: based on the SCORM Technical Working Group meeting, added these two requirements to clearly define the LMS behavior for evaluation of the completion status data model element. • REQ_60.3: Added a clarification note to indicate the expected behavior of an LMS if the <adlcp:completionThreshold> is not

Version	Release Date	Description of Change
		provided in a manifest. • REQ_62.3.1 and REQ_62.3.2: added an ADL Note add some explanatory support to the requirements. • REQ_63.4.3: based on the SCORM Technical Working Group meeting updated this requirement to change the Suspend All to an Exit All navigation request. • REQ_63.5: added this requirement to clearly define the LMS behavior for initializing the cmi.exit data between learner sessions. • REQ_64.3.5, REQ_64.5.2.6: add this requirement that defines the LMS behavior for handling of SetValue() calls when the identifier is not unique. • REQ_65.3: Added a clarification note to indicate the expected behavior of an LMS if the <adlcp:dataFromLMS> is not provided in a manifest. • REQ_70.3: Added a clarification note to indicate the expected behavior of an LMS if the <imsss:attemptAbsoluteDurationLimit> is not provided in a manifest. • REQ_72.1.3: added missing value, progress_measure, from list • REQ_72.3.3.2, REQ_72.3.3.3: updated language to assist in the understanding of which values are set. • REQ_72.7.5: changed cmi.interactions.n.id to cmi.objectives.n.id • REQ_74.3.3: added this requirement to clearly define the LMS behavior for initializing the cmi.scaled_passing_score when the satisfiedByMeasure attribute is set to false. • REQ_76.5: added this requirement to clearly define the LMS behavior for initializing the cmi.session_time data between learner sessions. • REQ_77.5, REQ_77.5.1, REQ_77.5.2 and REQ_77.5.3: based on the SCORM Technical Working Group meeting, updated these requirements to clearly define the LMS behavior for evaluation of the success status data model element. • REQ_77.5.4: based on the SCORM Technical Working Group meeting, added this one requirement to clearly define the LMS behavior for evaluation of the success status data model element. • REQ_78.2: based on the SCORM Technical Working Group meeting, changed the SPM for the cmi.suspend_data from 4000 characters to 64000 characters. • Added Section 2.1.7 User Interface Conformance Requirements. This section was added to ensure that a minimum set of User Interface requirements were being met by SCORM 2004 3rd Edition conforming LMSs. All SCORM 2004 3rd Edition LMSs are required to conform to these new requirements. Content Packaging Conformance Requirements Updates: • REQ_28.1.3: updated this requirement to add an ADL Note

Version	Release Date	Description of Change
		describing where the version number is defined. • REQ_28.6: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.5.3.1 and REQ_29.4.3.1: Updated requirment to change the expected value of this element to be 2004 3rd Edition. • REQ_30.5.4.1, REQ_29.4.4.1, REQ_29.6.3.8.1.1, REQ_29.6.3.8.1.2, REQ_29.6.3.9.4.1.1: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.6.3.6.2.2: based on the SCORM Technical Working Group meeting, added the ADL Note to recommend that (sub)manifest not be used. • REQ_30.6.3.6.2.4: added this requirement to clearly define that a leaf item shall have an identifierref. • REQ_30.6.3.6.4.2: based on the IMS Content Packaging maintenance work, added another parameter syntax that shall be supported • REQ_30.6.3.6.8.1.1: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.6.3.7.1.1: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.7.3.6, REQ_30.7.3.6.1, REQ_29.6.3.5.4 and REQ_29.6.3.6.1: Deleted requirements. The adlcp:persistState element has been deprecated for several releases. Removing the requirement with this release. • REQ_30.7.3.8.1.1: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.7.3.8.1.2: Updated requirment to remove notion of a SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.7.3.9.1, REQ_29.6.3.9.1: Updated requirement for clarity on what is meant by local files. • REQ_30.7.3.9.1.1: Updated requirement to indicate that a <file> element's href attribute shall be equivalent to the <resource> element's href attribute exclusive of any URL parameters. • REQ_30.7.3.9.4.1.1: Updated requirment to remove notion of a

Version	Release Date	Description of Change
		SCORM metadata application profile. Changed requirement to indicate that if metadata is provided it shall be well-formed and valided to its controlling document. • REQ_30.7.3.10.1.2: based on the IMS Content Packaging maintenance, added this requirement to clearly define that a dependency's identifierref must reference a resource that is in the current manifest. • REQ_30.8: updated this requirement to add an ADL Note recommending not to use (sub)manifests until further notice. • REQ_30.8.1, REQ_30.8.2, REQ_30.8.3, REQ_30.8.4, REQ_30.8.5, REQ_30.8.6 and REQ_30.8.7, based on the SCORM Technical Working Group meeting, removed the requirements for processing and handling (sub)manifests. • Due to a numbering and identification issue, updated Requirements 29.5 and all of it sub-requirements to 29.4. This caused several other requirements to change number scheme also. • REQ_31.6.1.1, REQ_31.6.2.1, REQ_31.6.3.1: Updated the multiplicity requirements from 0 or 1 to 1 and only 1 to match IMS Simple Sequencing XSD requirements. • REQ_31.6.1.1.2, REQ_31.6.2.1.2, REQ_31.6.3.1.2: Updated the multiplicity requirements from 0 or More to 1 or More to match IMS Simple Sequencing XSD requirements. • REQ_31.6.1.1.2.1.3: Added new requirement dealing with referencedObjective referencing an objectiveID of a <primaryObjective> or <objective> defined for the activity. • Added missing requirements REQ_31.6.2.1.2.1.2, REQ_31.6.2.1.2.1.3, REQ_31.6.3.1.2.1.2 and REQ_31.6.3.1.2.1.3.. Navigation and Presentation Extension Conformance Requirements • REQ_33.1.1.1: Updated requirement to include new vocabulary token of suspendAll, exitAll and abandonAll. SCO Conformance Requirements Updates: • Updated REQ_99.2 to add an ADL Note describing that the logout value is being deprecated and should not be used. • REQ_108.5.5 added new requirement for handling the cmi.objectives.n.id being set more than once. Metadata Conformance Requirements Updates: • Removed SCORM requirements for metadata elements. Sequencing Conformance Requirements (Appendix A) Updates: • Updated all figures to maintain a consistent look and feel between all of the SCORM documents. • Updated the use of Precondition to change it to Pre Condition to match the term used in the SCORM 2004 Sequencing and Navigation book.

Version	Release Date	Description of Change
		• Updated the Sequencing Information for Activity 4 in Test Case RU-7a. Changed Post Condition Sequencing Rule to a Pre Condition Rule. • Updated Step 5 and Step 9 in the Test Script for Test Case RU-7c. Changed cmi.progress_status to cmi.completion_status. • Updated the language for Activity 2 and ADL Note found in Test Case SX-3. Removed the note stating that Activity 2 was a (sub)manifest. The corresponding test package in the Conformance Test Suite has been updated to remove the (sub)manifest as well. • Test Case SX-4a was added to test LMS support for handling of an Abandon Navigation Request and to ensure that the run-time data is not persisted when an Abandon Navigation Request is processed. • Test Case SX-4b was added to test LMS support for handling of an Abandon All Navigation Request and to ensure that the run-time data is not persisted when an Abandon All Navigation Request is processed. • Test Case CM-2b: Added new secondary objective obj3 to the list of objectives for Activity 1 to support run-time initialization tests. Added an empty primary objective and one secondary objective to Activity 3 to support run-time initialization tests. • Test Case CM-3a: Added an empty primary objective and one secondary objective to Activty 2 to support run-time initialization tests. Added an empty primary objective and 4 secondary objectives to Activity 3 to support run-time initialization tests. • Test Case CM-3b: Added a primary objective (PRIMARYOBJ) and one secondary objective (obj1) to Activity 2 to support run-time initialization tests. Added a primary objective (PRIMARYOBJ) and 5 secondary objectives to Activity 3 to support run-time initialization tests. • Test Case CM-4b: New test case was added to test that an LMS can initiate a sequencing session through the learner selecting a non-leaf activity in the activity tree. • Test Case CM-4c: New test case was added to test that an LMS can initiate a sequencing session through the learner selecting a non-leaf activity in the activity tree. • Test Case CM-4d: New test case was added to test that an LMS can choose “invisible” activities and is capable of launching the base case involving Choice Navigation Requests. • Test Case CM-6: A new test case was added to test LMSs behavior in situations where “walking off the activity tree” may occur. For example, this would cover cases where an Continue Navigation Request would traverse past the last logical activity in the Activity Tree and end the Sequencing Session. • Test Case CM-7a, CM-7b, CM-7c, CM-7d and CM-7e: New test cases were added to test that LMSs correctly handle User Interface interoperability requirements. • Test Case CM-8: New test case was added to test LMSs only

Version	Release Date	Description of Change
		"write" shared global data when their attempt ends, not when local objective state is evaluated during a tree traversal. • Test Case RU-11: Added an empty primary objective to Activity 3 to support run-time initialization tests. • Test Case RU-12a, Test Case RU-12b: New test cases were added to test that LMSs only modify local tracking information as a result of a successful Rollup Rule evaluation. • Test Case SX-5: A new test case was added to test LMSs behavior in situations where "walking off the activity tree" may occur. For example, this would cover cases where an Continue Navigation Request would traverse past the last logical activity in the Activity Tree and end the Sequencing Session. • Test Case SX-6: New test case was added to test that an LMS correctly maps a cmi.completion_status of "not attempted" to the activity tracking state of "incomplete". • Test Cases SX-7a, SX-7b, SX-7c, SX-7d and SX-7e: New test cases were added to test evaluation state data for re-try. • Test Case OB-3b: Added a test to the test case that ensures an LMS properly reset Shared Global Objectives across multiple attempts on the Activity Tree. • Test Case OB-3c: Added a primary objective (PRIMARYOBJ) to Activity 8 to support run-time initialization tests. Updated the test case to ensure that LMSs only "write" shared global data when their attempt ends, not when local objective state is evaluated during a tree traversal. • Test Case OB-7a, Test Case OB-7b: New test cases were added to test the initialization of the objectives collection from shared global state and data mapping from the objectives collection to the primary objective. • Test Case OB-8a, Test Case OB-8b: New test cases were added to test that an LMS properly processes an Exit Parent rule that also ends the Sequencing Session by exiting the root of the Activity Tree. • Test Case OB-9a, Test Case OB-9b: New test cases were added to test persistence of shared global objectives across "courses". • Test Cases OB-10a, OB-10b, OB-10c, OB-10d, OB-11a and OB-11b were added to test the initialization of the objectives collection from shared global state on untraced activities. These tests also test that the data mapping from the SCO's cmi.objectives are not applied to untracked activities. • Test Cases OB-12a, OB-12b, OB-12c, OB-13a, OB-13b and OB-13c: New test cases were added to test that an LMS considers Objectives Measure Status separate from Objective Progress Status. • Test Case OB-14a: New test case was added to test that LMS shared global oabectives initialization and threshold behaviors are being applied consistently; that sub-seconds are added correctly between learner sessions; and that LMS run-time behavior regarding application of satisfaction thresholds to secondary

Version	Release Date	Description of Change
		objectives are being handled correctly. • Test Case OB-14b; New test case was added to test the management of cmi.objectives in conjunction with initialized objectives. • Test Case OB-15; New test case was added to test that and LMSs initialize Attempt Progress Information to “unknown” for activites that have not been attempted and that have not been involved in the Extended Rollup Process. • Test Case T1a, T1b: The documented test case T1 was split into two documented test cases T-1a and T-1b. Global objective identifiers were updated to reflect the split.