



**Army Business Rules for
SCORM® 2004 3rd Edition
Conformant Courseware**

Version 1.1.1

30 September 2014

SUMMARY of CHANGE

Business Rule Version	Release Date	Description of Change
V1.0	30 June 2014	Initial Release
V1.1.0	7 July 2014	Inserted Army Business Rule C2 - Question Information Processing (Section 6.2)
V1.1.1	30 September 2014	SCORM® interaction data element requirements updated.

Army Business Rules for SCORM® 2004 Conformant Courseware

1. Executive Summary	6
1.1 Notations Used.....	7
1.2 List of Army Business Rules.....	7
1.2.1 Army Instructional Business Rules.....	10
1.2.2 Army Metadata Business Rules.....	11
1.2.3 Army Filename Specifications.....	12
1.2.4 Army Scripting and Coding Business Rules.....	13
1.2.5 Army SCORM® Business Rules.....	14
1.2.6 Army Delivery Business Rules.....	16
2. Old Business Rules Compared with the New Business Rules	17
2.1 Army Conformance and Validation Procedures for the Old Business Rules	17
2.2 Map Old Business Rules to New Business Rules	20
2.3 Comparing Army Validation of the Old Business Rules to the New Business Rules ...	22
2.4 Evaluating the new set of Army Business Rules.....	27
3. Army Instructional Business Rules	31
3.1 SCORM® 101.....	31
3.2 SCORM® and Instructional Design	32
3.3 SCORM® Concepts and Processes.....	32
3.3.1 Reusability and Independent Learning Objects	32
3.3.2 The "Page"	32
3.3.3 SCORM® Content.....	33
3.3.4 Army Classification of SCOs.....	34
3.4 The Army Instructional Content Business Rules	36
3.5 SCO Titles	36
3.5.1 SCO Example.....	38
3.5.2 Army Independent LCO “Essential” Instructional File Placement	39
4. Army Metadata Business Rules	40
4.1 XML Binding for Separate Metadata Files	40
4.2 SCORM® Metadata Schema Version.....	40
4.3 Relative Path Example	41
4.4 Content Organization Metadata Referenced on the Manifest	42
4.5 SCO Metadata Referenced on the Manifest.....	42
4.6 Army Metadata Schema Business Rule	42
4.7 Setting up SCORM® Metadata Schemas	43
4.8 Army Metadata Requirements	44

4.9	Developing Army required SCORM® Metadata Files.....	44
4.9.1	Catalog and Entry Identifier.....	46
4.9.2	Title of Learning Resource	46
4.9.3	Language of Learning Resource	46
4.9.4	Description of Learning Resource	47
4.9.5	Keywords	48
4.9.6	Type of Metadata	49
4.9.7	Version of Learning Resource	49
4.9.8	Status of Package Submittal.....	50
4.9.9	Proponent's Role	50
4.9.10	Proponent's Name and Address	51
4.9.11	Date of Submittal	51
4.10	Meta-Metadata Catalog and Entry Identifier	52
4.10.1	Meta-Metadata Schema	52
4.10.2	Language of the Metadata File	53
4.10.3	File Formats	53
4.10.4	Cost of Learning Resource.....	54
4.10.5	Copyright and Other Restrictions	54
4.10.6	Classification.....	55
5.	Army Filename Characters and File Reference Business Rule.....	59
5.1	File Naming Convention	59
6.	Army Scripting and Coding Business Rules	61
6.1	Bookmarking.....	61
6.1.1	Government Validation of the Bookmarking Business Rule.....	61
6.1.2	Bookmarking Sample JavaScript.....	61
6.2	Question and Answer Processing.....	62
6.3	Communication with the LMS (Run-Time Environment) Examples	63
6.3.1	Begin Communication	65
6.3.2	End Communication	66
6.3.3	Army Business Rule C3 Code Exit.....	66
6.3.4	Examples of Properly Coding SCO Exits.....	67
6.3.5	Exit Status Example.....	71
7.	Army SCORM® Business Rules	72
7.1	Global Objectives	72
7.2	Screenshot of Manifest File.....	73
7.3	Manifest in Detail.....	75
7.4	Army SCORM® Business Rule S2 Item Analysis	78
7.5	Session Time Example.....	78

7.6	Total Time Example.....	81
7.7	Overview	81
7.7.1	SCORM® Required Interaction Data Fields	82
7.7.2	Army Required SCORM® Interaction Data Fields.....	84
7.7.3	Test Item Data Collection Example.....	90
7.7.4	Total Time Example	91
7.7.5	Mastery Score Example	91
7.7.6	Learner Scoring and Success Status	92
7.7.7	Initializing Learner's Success Status.....	92
7.7.8	Confusing Success Status with Completion Status.....	94
7.7.9	Completion Status Example.....	94
7.7.10	GetLastError	95
7.7.11	Scaled Passing Score Example	95
7.7.12	Army Validation Business Rules	96
8.	Army Courseware Delivery Business Rules.....	99
8.1	Packaging and Delivery	99
8.1.1	Creating the Manifest File	100
8.1.2	How to Create a Content Package per the SCORM® Implementation Guide	101
8.1.3	Final Courseware Packaging and Delivery Requirements.....	103
8.1.4	Army Delivery Documentation Business Rule.....	104
9.	APPENDIX A: Test Item Data (Interactions) Table.....	105
10.	APPENDIX B: Metadata Tables.....	107
11.	APPENDIX C: SCORM® Data Types	110
12.	APPENDIX D: Glossary and Acronyms	113

1. Executive Summary

This document is intended for use by personnel involved with Army activities developing Distributed Learning (DL) content, both government employees and private company employees. Army activities developing DL includes all individuals and groups involved in the design, development, and programming of training to be submitted to the Army for hosting by Army Learning and Content Management Capabilities (ALCMC). Programmers from contractors and/or proponent agencies designing and developing DL courseware for the Army must thoroughly understand this document.

These Business Rules apply to all Army DL, with emphasis on tracked, credit producing Web-based training. The Army has adopted the Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition specification to support the interoperability of content and to supply a consistent data communication model between content and the hosting Learning Management System (LMS). All credit producing, tracked Army DL courseware will be served on a SCORM® 2004 conformant LMS. All Business Rules must be adhered to by Army DL content producing activities when developing courseware to be hosted by ALCMC. As of this publication, the major ALCMC LMSs are the Army Learning Management System (ALMS) with Saba enterprise LMS hosting courseware, and Enterprise Lifelong Learning Center (ELLC) with Blackboard Learning System hosting courseware. Examples are included to assist in the development of SCORM® 2004 conformant courseware and to aid in consistency.

The Advanced Distributed Learning (ADL) Technical Working Group (TWG) acknowledges that the SCORM® certification process for LMSs is inadequate to fully support total and worry free interoperability of all SCORM® content between different LMSs. There is no guarantee that a SCORM® Content Package will behave in the same manner across different ADL-certified LMSs. Therefore, your courseware **must** be tested in your target LMS. Note that decreasing SCORM® Content Package complexity by limiting the sequencing required and unique rollup requirements increases the content package's durability, maintainability and portability – and therefore reusability. Unfortunately, simple SCORM® Content Packages are the only packages that may work between wide varieties of SCORM® conformant LMSs.

1.1 Notations Used

The following notations are used throughout the document and explained below:

-  **Army Business Rule (Army):** An Army mandatory requirement for implementing the SCORM® specification. These requirements must be met for SCORM® 2004 courseware to be accepted by the Army.
-  **Best Practice:** Recommended SCORM® 2004 practice for what works best based on lessons learned. Alternative approaches are acceptable as long as they comply with SCORM® 2004 and Army SCORM® 2004 requirements and can be implemented in the LMS appropriate for the courseware.
-  **To Be Developed (TBD):** Certain portions of this document, or material referenced in this document, are still in development. These sections will be updated in future iterations of this document.
- **PROGRAMMER INFO:** Programming information separated from general guidance to emphasize and isolate important points.

This document provides Army Business Rules, definitions, explanations, and sample code for developing Army DL destined for fielding in the ALCMC. **The Army Business Rules are mandatory requirements for all Army DL courseware and provide normative information concerning the design, development, delivery and other fundamentals validated during US Army Acceptance review and Functional Testing.** All script snippets and sample script tables are informational in nature, since each line of script only has full context when part of an executing collection. However each snippet is intended to be functional if duplicated and placed in an actual DL content package.

Changes to this document are anticipated to coincide with updates to the SCORM® specification, ALCMC updates, Technology evolution, changing Army Training and DL strategies, as well as through Army DL community feedback, and the identification of lessons learned.

1.2 List of Army Business Rules

This section is informational to provide SCORM® and Army DL fundamentals before presenting the set of Army Business Rules.

The new list of Army Business Rules changes the total list of Business Rules from 20 to 19, and introduces six groups:

- Instructional (3)

- Metadata (3)
- Filename Characters (1)
- Scripting and Coding (2)
- SCORM® (6)
- Delivery (3)

Introductory terms: Three important terms need to be introduced and described before introducing the list of Army Business Rules:

Page: The term page has long been used to describe eLearning content presentation. Even though this document refers to the page paradigm frequently, in nearly all cases, any other computer presented “item” can replace the term page. For instance “press next for the next *topic*”, or “every *model* of the Binford widget must be completed before accessing any *model* of the Westinghouse widget.”

SCO: The SCORM® term for a small chunk of learning material is Sharable Content Object (SCO). In this document a SCO is not sharable or reusable because it is a SCO. This document will use sharable and reusable interchangeably, and modified and repurposed interchangeably. SCOs are only sharable if they are designed and developed to be shared and reusable. SCOs are no different than any other eLearning collection or chunk; what they are made of is what they are, in other words:

If a SCO’s content is created following rules which support total and worry free reuse then that SCO is reusable.

Reusable Objects: The same reuse dialogue presented for SCO information is presentable for other learning content objects or chunks. The Business Rules are written for SCORM® content; however, in the body of this document it shall be noted that SCO represents learning content objects and chunks in general and not just SCORM® learning content objects. This position reflects the recognition that eLearning content objects, SCORM® conformant or not, can be arbitrarily sized and created with traits similar to SCOs and content packages, that is, all content objects can be independent and context-neutral content.

Why SCORM®? What does SCORM® provide? The previous paragraph makes the point the any eLearning content object can be created ready to be reused and SCOs are not reusable unless they are designed and developed to be reusable. In other words, eLearning content objects created following reusable content design rules **can** be reused on other content hosting platforms. Take notice that the power of SCORM® is that the previous claim is truly accurate when changed to read: **Non-scoring, non-progress tracking, non-LMS communicating** eLearning content objects created following reusable content design rules can be reused on other content hosting platforms. Addressing Learner tracking, the following is true with respect to “reuse without SCORM®”; reuse of Learner progress tracking eLearning content created following reusable content design rules is **restricted to the same vendor’s compatible LMSs**. With

SCORM® introduced, the following becomes true; using SCORM® specifications to create and implement learner progress tracking eLearning content object can allow the objects to be **reused in any SCORM® conforming LMS environment**, including other vendor’s LMS products supporting the same SCORM® version.

A FINAL IMPORTANT NOTE

The last and most significant information for all Army DL developers providing content to be fielded in the Army DL Learning Content Management Capability (ALCMC) is how to use the following four critical resources:

1. SCORM® 2004 3rd Edition reference suite: This three volume set must be examined and remain available for reviewing. There is no better source to get syntax and detailed information about the SCORM® small command set, strict data model elements, and the “anything but simple” SCORM® implementation of Sequencing, Navigation and Rollup rules. SCORM® 2004 3rd Edition reference set begins with three manuals: 1) Content Aggregation Model (CAM), 2) Run-Time Environment (RTE), and 3) Sequencing and Navigation (SS&N). These three volumes are available from the ADL website in the *SCORM 2004 3rd Edition Technical Documentation* area. A link is not provided here to encourage finding supporting material relevant to design and development of good SCORM® conformant content available from the ADL website, <http://www.adlnet.org/> .
2. This book: Army Business Rules for Army SCORM® 2004 3rd Edition Conformant Courseware, is the best guide for getting your submitted content to operate properly when fielded on the ALCMC by the Army.
3. The Advanced Distributed Learning Initiative (ADL) document “[SCORM Users Guide for Programmers](#)”: This guide is to help the eLearning programmer become familiar with SCORM® 2004, this guide explains different manifest tags and how they interact with the SCORM® data model elements using multiple examples. This guide is the best instruction available to help you get your submitted content to pass Individual Trials, Group Trials, AND Functional Testing, all of which are essential to getting your content fielded in the ALCMC. This guide promotes several SCORM® 2004 4th Edition features, so be cautious to not focus or rely on the clearly marked SCORM® 2004 4th Edition features, since all Army DL content is limited to SCORM® 2004 3rd Edition.
4. Army Acceptance Criteria for Combined Arms Products for Distributed Learning (CAPDL) Courseware is the most important guide for getting your submitted content to be accepted by the Army, the link for that document is on this page [SCORM® Acceptance Criteria](#).

The Army Business Rules are normative information. The Army Business Rules apply to all Army DL; tracked, credit producing and non-tracked, non-credit producing DL is addressed by one set of Army Business Rules.

1.2.1 Army Instructional Business Rules

Army Business Rule I1 (Instructional Content Rules): Independent Learning Content Objects (LCOs) shall be able to stand alone as discrete blocks of learning content. The learning content of an independent LCO shall not contain references (hyperlinks, narration, etc.) to another Content Object or aggregation. Independent LCO content and titles must not include sequencing numbers or other hierarchical level instruction titles or words (course, phase, module, lesson, etc.). If an independent LCO contains an Enabling Learning Object (ELO), "ELO" shall not be referenced within the content or title of the LCO. Independent LCO titles must not include the Military Occupational Specialty (MOS), Skill Level, Skill Qualification Identifier (SQI), or Additional Skill Identifier (ASI). Independent LCO content shall not contain the acronyms "SCO", "TLO", or "ELO"; or references to a course map or include transitional statements such as "The next lesson will explain..."

Tracked Independent Learning Content Objects (LCOs) are SCOs and Assignable Units in SCORM® and Aviation Industry Computer-Based Training Committee (AICC) Distributed Learning specifications, respectively. Untracked Independent Learning Objects may be composed of web pages or other Interactive Multimedia Instruction (IMI) presentations that stand alone as discrete blocks of learning content.

Army Business Rule I2 (Instructional Titles): All Learning Content Objects (LCOs) and Sharable Content Objects (SCOs), both dependent and independent, shall display the same topic title as a Hyperlink in the Table of Contents (or Menu) and in the instructional content presented to the learner.

Army Business Rule I3 (Instructional Learning): Instructional content that directly contributes to satisfying mastery requirements must be physically located within the LCO content package (within the root folder or subfolder) and not referenced by a URL. This is true for all Army DL learning content, dependent or independent that is loaded into the ALCMC.

PROGRAMMER INFO: The government may allow exceptions for non-critical material, usually for large manuals such as Soldiers Manual of Common Tasks, Field Manual, Soldier Training Publication, and so on. All files residing external an LCO package shall be protected from removal, replacement, or updates by being stored in an Army repository or within an Army LMS. URLs for these external files shall be fully qualified.

1.2.2 Army Metadata Business Rules

Army Business Rule M1 (Metadata External): Metadata must be in separate eXtensible Markup Language (XML) files and referenced from within the manifest file using the XML binding for separate metadata file references.

Army Business Rule M2 (Metadata Schema): The metadata schemas that must reside in the root folder must also be located in the folder(s) where the metadata resides. Folders that contain metadata shall also have the metadata schemas and must be copied into this folder.

Army Business Rule M3 (Metadata Required):

SCORM® content: Metadata is required for ALL Content Packages and Resource Packages, all manifest Organization level items including each SCO and all launchable assets.

Non-launchable Assets and resource aggregations must have metadata only upon the government's request.

Non-SCORM® content: Learning Content Objects that are not required to follow SCORM® communication specifications have the same metadata requirements. The government may specifically identify Learning Content Objects as exempt from Metadata requirements.

All metadata must be formatted in accordance with the instructions in section 4, Army Metadata Business Rules, and Appendix B: Metadata Tables.

1.2.3 Army Filename Specifications

Army Business Rule F1 (Filename Characters): The Army file and folder naming requirements follow the Network Working Group Request for Comments 3986, commonly referred to as RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax.

Forbidden Characters:

The two delimiter character sets, Generic and Subcomponent, make up the first group of characters the Army does not allow in file or folder names. These characters are presented below using a vertical bar “|” delimiter:

“.” | “/” | “?” | “#” | “[“ | “]” | “@” | “!” | “\$” | “(“ | “)”” | “*” | “+” | “,” | “;” | “=” | ’ (right single quote)

Additional forbidden characters the Army forbids to be in file or folder names are characters found or reported to cause problems in a UNIX® Operating System Environment. These characters are presented below using a comma, “,” delimiter:

|, <, >, \, ”, ` , ~, {, }, &, ^, <space>, <tab>

External resource names, which contain forbidden characters, shall be referenced in Army DL content with a URL that percent encodes each forbidden character.

Allowable Characters:

Allowable characters in Army DL file and folder names are part of the RFC 3986 unreserved character set: alpha (lowercase only) | digit | “-“ | “_” | “.”

The full list characters the Army allows in file and folder names:

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0,1,2,3,4,5,6,7,8,9, -, _, . (period)

Additional Army File and Folder Naming Guidance:

- A single period shall be used to separate the file name from its extension. Folder names shall not contain a period.
- Long file names are allowed but must conform to ISO 9660 with Microsoft Joliet extension and the Army's file naming scheme. Joliet allows filenames up to 64 characters in length. Joliet allows spaces but the Army's naming scheme does not.

Subcomponent delimiter characters (not allowed) = “!” | “\$” | “”” | “(“ | “)”” | “*” | “+” | “,” | “;” | “=”

1.2.4 Army Scripting and Coding Business Rules

Army Business Rule C1 (Code – Bookmarking): Sharable Content Objects (SCOs) and other learner progress tracking Learning Content Objects (LCOs) shall provide each learner the ability to bookmark the learner’s progress whenever the learner exits prematurely. SCOs and LCOs must allow the learner the option of resuming progress in that SCO or LCO from the bookmarked location posted during the previous learning session. Standard exceptions to the Bookmarking requirement exist, such as credit producing assessments, short learning objects with a single entry point, or other short learning objects such as summaries and introductions. Proponents may designate specific Bookmarking processing requirements, and proponents may also designate any SCO or LCO exempt from the Bookmarking requirement.

Army Business Rule C2 (Answer Data Processing): Assessment content packages must not contain question and answer data in a format clearly recognizable by the learner. Question and answer data must be altered, or obfuscated, into a format not easily identifiable and readable by the learner:

- When the assessment process includes compiling or publishing, question and answer data must be embedded within a compiled or published file; such as a Flash '.swf' file.
- When question and answer data is in a text data source such as a JavaScript file or xml file, the text files must be obfuscated using a binary to text encoding method.

Army Business Rule C3 (Code Exit): Regardless of learner SCO exit method, learner progress shall be recorded consistently and the SCO must terminate communication with the LMS (for example, a SCO navigation request, LMS navigation button, closing the browser window, pushing an ‘Exit’ button on the SCO page, etc.).

1.2.5 Army SCORM® Business Rules

Army Business Rule S1 (SCORM® Objectives): The scope of all manifest global objectives shall be the manifest itself. Manifest global objectives shall be shared within a single SCORM® imsmanifest.xml file. SCORM® sets the default scope for global objectives to "true". This rule requires that the organization attribute objectivesGlobalToSystem be set to "false" in all DL content submitted to the Army. For example, a valid organization tag and attributes entry is <organization identifier="37W40 examinations" structure="hierarchical" adlseq:objectivesGlobalToSystem="false">.

Army Business Rule S2 (SCORM® Item Analysis): Developers of SCO assessments must make use of the SCORM® Interactions Data Model Element to record information about the learner's response for validation purposes. Interactions Data Model Elements Identifier, Type, Correct Responses, Latency, Learner Response, Result, and Timestamp are required for each question. The Interaction element Weighting Interaction element is required for an examination that varies the scoring weight for questions. Figure 7.9a presents the Army requirements for each SCORM® Interaction Data Model Element.

Army Business Rule S3 (SCORM® Pass): Graded assessment Sharable Content Objects (SCOs) shall ensure that an appropriate value for the success_status, either 'passed' or 'failed' is set in the LMS. Graded assessment SCOs that pass a learner score to an LMS shall declare their mastery score to the LMS using the manifest element <imsss:minNormalizedMeasure>.

Furthermore:

- These SCOs shall use the mastery score retrieved from the LMS when reporting the passing grade to the learner.
- These SCOs shall not set the satisfiedByMeasure attribute of their corresponding <imsss:primaryObjective> element to "false".

Army Business Rule S4 (SCORM® Methods and Data Models Elements Required for Instructional Content – Non-scoring SCOs): The Army requires three methods be called by every SCO:

- Initialize()
- Terminate()
- Commit()

Also, the Army requires the following SCORM® data model elements be either set for all non-exam SCOs:

- cmi.exit
- cmi.success_status
- cmi.completion_status
- cmi.location
- cmi.session_time

Method calls and data element usage are validated during the Army Multi-log Parser processing of the Run-time Environment log file for exam SCOs.

Army Business Rule S5 (SCORM® Methods and Data Models Elements Required for Performance Measure Content – Scoring SCOs): The Army requires four methods be called by every scoring SCO:

- Initialize()
- Terminate()
- GetLastError()*
- Commit()

Also, the Army requires the following SCORM® data model elements be set for all scoring SCOs:

- cmi.exit
- cmi.completion_status
- cmi.scaled_score
- cmi.session_time
- cmi.interactions.n.correct.response.n.pattern
- cmi.interactions.n.learner_response
- cmi.interactions.n.latency
- cmi.interactions.n.timestamp
- cmi.interactions.n.type
- cmi.interactions.n.result
- cmi.interactions.n.weighting (when question is weighted differently than others in SCO)

Additionally, the SCORM® data model element cmi.success_status and cmi.scaled_passing_score value must be read via the GetValue method for all scoring SCOs. Method calls and data element usage are validated during the Army Multi-log Parser processing of the Run-time Environment log file for exam SCOs.

* SCORM® GetLastError() method requirements are clarified in Army Business Rule S6

Army Business Rule S6 (SCORM® Error Management): The SCORM® API has three API methods used to handle errors, none affect or alter the error code for the API Instance:

- GetLastError()
- GetErrorString()
- GetDiagnostic()

The Army requires that the GetLastError() method be called from all scoring SCOs and this is validated during the Army Multi-log Parser processing of the Run-time Environment log file for a scoring SCO.

1.2.6 Army Delivery Business Rules

Army Business Rule D1 (Delivery Rules): All SCOs (content and graded assessments) must be contained in a content package. SCORM® Content Packages must contain a manifest (imsmanifest.xml) file and all of the SCORM® and extension schemas in the root of the package.

All physical files required for the courseware must be referenced locally and contained within the content package and disclosed on the imsmanifest.xml file. Metadata is supplied as separate files and these files must be contained within the content package and disclosed on the manifest file.

Army Business Rule D2 (Delivery Consumables): The Government requires a “Gold Copy” of all eLearning content to be delivered using SCORM® content packaging guidelines. The “Gold Copy” requirement includes all file-based content that is consumable by the learner. Furthermore, the files will be packaged into logical bundles and delivered as Package Interchange File (PIF) file(s). Other contract deliverables (for example, original source files, SCORM® testing log files, answer keys, loading instructions, etc.) must be delivered as separate files external to the courseware PIF file(s).

Army Business Rule D3 (Delivery Documentation): All SCORM® content submitted to the Army to be hosted on an Army Learning Management System must be accompanied by verification of quality documentation in two areas: 1) Screen captures showing LMS display of Learner progress to completion, normally successful completion and unsuccessful completion for graded examinations, and 2) Each SCORM® Content Package must have SCORM® Certification Test Suite (CTS) log files for the Content Package conformance test option and each SCO must have the log file from the Run-Time conformance test option.

Figure 1.1a

2. Old Business Rules Compared with the New Business Rules

The information in this section, section #4 is informational.

The business rules in this document are the result of examination and discussions that reconstructed the entire existing set of business rules. The goal is to increase the value of conforming to the new Business Rules, provide a more direct specification of Army DL requirements, and to provide direct explanations of Army DL testing and acceptance, where the Army validates submitted DL courseware meets “Army mandatory” rules.

The business rules in this document are the result of examination and discussions that reconstructed the entire existing set of business rules. The goal is to increase the value of conforming to the new Business Rules and to provide a direct specification of Army DL requirements, in addition to direct explanations of Army DL acceptance testing and mandatory rules, in which the Army applies when examining submitted DL courseware.

2.1 Army Conformance and Validation Procedures for the Old Business Rules

The first table presents the previous Army Business Rule set and Army conformance testing for each.

Old Business Rule	Keep or Remove?	Does Army Test?	Is Army Test Automated?	Is Army Test Manual?
#01 No course-to-course hyperlinks	<i>Keep</i>	Yes	No	Yes
#02 Content Matches TOC Title	<i>Keep</i>	No	No	No
#03 Internal Context only; no sequence or hierarchical references	<i>Keep</i>	No	No	No
#04 Must contain all critical instructional material	<i>Keep</i>	No	No	No
#05 Only an LMS can launch a SCO	<i>Remove</i>	Yes	No	Yes (LMS enforced)
#06 File naming conventions	<i>Remove</i>	No	No	No
#07 Illegal filename characters	<i>Keep</i>	No	No	No
#08 URLs must use escaped character codes	<i>Keep</i>	No	No	No
#09 Help required for internal navigation	<i>Remove</i>	Yes	No	Yes
#10 Must bookmark learner progress	<i>Keep</i>	Yes	Yes	Yes
#11 SCORM® manifest objectives scope is local, not system	<i>Keep</i>	No	No	Yes (assumed)*
#12 Assessment must post Army mandatory test item elements	<i>Keep</i>	Yes	Yes	Yes
#13 Metadata required for packages, aggregations, and content objects	<i>Keep</i>	Yes	Yes	No
#14 No inline metadata	<i>Keep</i>	No	No	No
#15 Schemas in Root and folders with metadata files is required for test tools to operate properly.	<i>Keep</i>	Yes	Yes	No
#16 Graceful Exits: All Content Objects communicate termination data for all possible learner selected exit paths	<i>Keep</i>	Yes	No	Yes (not enforced)
#17 SCORM® assessments must fully use an imsmanifest.xml tag for minimum mastery score	<i>Keep</i>	No	No	Yes (assumed)*
#18 Learner's credit earned for Learning Content Objects will not be rescinded	<i>Remove</i>	No	No	No
#19 Packages - Full disclosure	<i>Keep</i>	Yes	Yes	No
#20 Packages - Consumables	<i>Keep</i>	NA	NA	NA

Table 2.1

* The Global Objectives scope and the processing of the manifest tag for minimum passing score is assumed to be conformant if Functional Testing doesn't find errors in repeated registrations and assessments work properly.

Army Validation of Old Business Rule Set: Nine (9) of the twenty (20) old Business Rules are validated by Army examination for conformance in submitted Army training. Five are tested by automated Army test suite tools. Eight are either manually validated by Army examination or validated by inference when passing Functional Testing.

2.2 Map Old Business Rules to New Business Rules

The second table shows how we have mapped the old Business Rules into the new Business Rules:

Old Business Rule	Old Business Rule <small>(Short Name)</small> / New Business Rule	Old Valid-ated?	New Valid-ated?
#01 No course-to-course hyperlinks	#01 Hyperlink scope Instructional Rules - I1 Rules	No	Yes
#03 Internal Context only; no sequence or hierarchical references	#03 Local context	No	
#02 Content Matches TOC Title	#02 Titles match Instructional Rules - I2 Titles	No	Yes
#04 Must contain all critical instructional material	#04 All instruction Instructional Rules – I3 Teaching	No	Yes
#14 No inline metadata	#13 Metadata Required Metadata Rules - M1 External	Yes	Yes
#15 Schemas in Root and in folders with metadata files is required for test tools to operate properly	#14 Metadata in files Metadata Rules - M2 Schemas	No	Yes
.#13 Metadata required for packages, aggregations, and content objects	#15 Metadata Schema Metadata Rules - M3 Required	Yes	Yes
#08 URLs must use escaped character codes	#08 Escape Characters Filename Rules - F1 Characters	No	Yes
#07 Illegal filename characters	#07 Bad file Characters	No	

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Valid- ated?	New Valid- ated?
#11 SCORM® manifest objectives scope is local, not system (LMS)	#11 Manifest Objective tag SCORM® Rules – S1 Objectives	No	Yes
#12 Assessment must post Army mandatory test item elements	#12 Item Analysis SCORM® Rules – S2 Item Analysis	Yes	Yes
#17 SCORM® assessments must fully use an imsmanifest.xml tag for minimum mastery score	#17Manifest Score SCORM® Rules – S3 Pass	No	Yes
	SCORM® Rules – S4 Methods and Data for non-scoring SCOs		Yes
	SCORM® Rules – S5 Methods and Data for Scoring SCOs		Yes
	SCORM® Rules – S6 Error Handling for Scoring SCOs		Yes
#10 Must bookmark learner progress	#10 Bookmark Code Rules – C1 Bookmarking	Yes	Yes
	Code Rules - C2 Answer Data Processing		Yes
#16 Graceful Exits: All Content Objects communicate termination data for all possible learner selected exit paths	#16 Easy Exit Code Rules – C3 Exit	Yes	Yes

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Valid- ated?	New Valid- ated?
#19 Packages - Full disclosure	#19 Package-Rules Delivery Rules - D1 Rules	Yes	Yes
#20 Packages - Consumables	#20 Package-Consumables Delivery Consumables - D2	NA	Yes
	Delivery Documentation - D3 1. Screen Shots of Learner Completion Paths 2. Certification Test Suite log files showing Army Mandatory SCORM® data calls for content and test SCOs.	NA	Yes
Removed Business Rules:			
#05 Only an LMS can launch a SCO	#05 LMS control	Yes	
#06 File naming conventions	#06 Name Conventions	No	
#09 Help required for internal navigation	#09 Navigation Help	Yes	
#18 Learner's credit earned for Learning Content Objects will not be rescinded	#18 Credit stays	No	

Table 2.2

2.3 Comparing Army Validation of the Old Business Rules to the New Business Rules

This table highlights the changes from the old Business Rules set compared to the new Business Rule set.

Grouping the business rules into six topics was the first change. These rules are presented in a generalized chronological order based on the Army DL Life Cycle Process:

1. Instructional Rules
2. Metadata Rules
3. Filename Rule
4. Coding Rules
5. SCORM® Rules
6. Delivery Rules

Instructional Rules – the content of these rules have remained similar, however the new rules are validated during the Design Reviews during Content Validation – occurring during the ADDIE Design Phase. This makes the Instructional Business Rules unique, since Army validation occurs before the instructional content is developed into the learner presentation, and well before the final submission packages are prepared for shipping to the government.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#01 No course-to-course hyperlinks	#01Hyperlink scope Instructional Rules - I1 Rules	Manual Validation during Functional Testing	Manually validated during content validation process in Design phase, usually before the content has been transferred into learner presentation format.
#03 Internal Context only; no sequence or hierarchical references	#03Local context	Not Validated	
#02 Content Matches TOC Title	#02Titles match Instructional Rules - I2 Titles	Not Validated	
#04 Must contain all critical instructional material	#04All instruction Instructional Rules – I3 Teaching	Not Validated	

Table 2.3a

Metadata Rules – these rules are not changed from the previous rules other than updating the mandatory entries.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#13 Metadata required for packages, aggregations, and content objects	#13 Metadata Required Metadata Rules - M1 External	Automated Validation using Metadata Editor Batch Validation	Automated Validation using Metadata Editor Batch Validation
#14 No inline metadata	#14 Metadata in files Metadata Rules - M2 Schemas	Not Validated	Automated Validation using Metadata Editor Batch Validation
#15 Schemas in Root and in folders with metadata files is required for test tools to operate properly.	#15 Metadata Schema Metadata Rules - M3 Required	Automated Validation using Metadata Editor Batch Validation	Automated Validation using Metadata Editor Batch Validation

Table 2.3b

Filename Rule – the old multiple filename rules have been combined into one presentation. While the old process requiring forbidden characters be entered into URLs using their respective percent encodes, RFC 2793 is presented as the source of this guidance. The allowable characters are now limited to dash, under bar, lower case letters and digits.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#08 URLs must use escaped character codes	#08Escape Characters Filename Rules - F1 Characters	Not Validated	Automated Validation by Army Multi-Log Parser
#07 Illegal filename characters	#07Bad file Characters	Not Validated	Automated Validation by Army Multi-Log Parser

Table 2.3c

Coding Rules – The Bookmarking Business Rule now clearly states the Army expectation that instructional content objects will set the cmi.location SCORM® data model element to the learner’s location for bookmarking purposes.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#10 Must bookmark learner progress	#10 Bookmark Code Rules – C1 Bookmarking	Automated Validation by Army Multi-Log Parser	Automated Validation by Army Multi-Log Parser
	Code Rules - C2 Answer Data Processing	Not Applicable	Validated in IMDP contents
#16 Graceful Exits: All Content Objects communicate termination data for all possible learner selected exit paths	#16 Easy Exit Code Rules – C3 Exit	Manual Validation during Functional Testing	Manual Validation during Functional Testing

Table 2.3d

SCORM® Rules – The SCORM® Business Rules gain three Business Rules that state the data model elements that the Army Multi-Log Parser verifies usage by examining ADL Certification Test Suite Run-Time Environment Option Log files

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#11 SCORM® manifest objectives scope is local, not system (LMS)	#11 Manifest Objective tag SCORM® Rules – S1 Objectives	Not Validated	Automated Validation by Army Multi-Log Parser
#12 Assessment must post Army mandatory test item elements	#12 Item Analysis SCORM® Rules – S2 Item Analysis	Automated Validation by Army Multi-Log Parser	Automated Validation by Army Multi-Log Parser
#17 SCORM® assessments must fully use an imsmanifest.xml tag for minimum mastery score	#17 Manifest Score SCORM® Rules – S3 Pass	Not Validated	Automated Validation by Army Multi-Log Parser
	SCORM® Rules – S4 Methods and Data for non-scoring SCOs	Not Applicable	Automated Validation by Army Multi-Log Parser
	SCORM® Rules – S5 Methods and Data for Scoring SCOs	Not Applicable	Automated Validation by Army Multi-Log Parser
	SCORM® Rules – S6 Error Handling for Scoring SCOs	Not Applicable	Automated Validation by Army Multi-Log Parser

Table 2.3e

Delivery Rules – Deliver rules gain a Business Rule, Delivery Documentation D3. This new Business rule states the requirement that each and every content package submitted to the Army for fielding on the ALCMC be accompanied by:

1. Screen shots showing LMS learner progress displays demonstrating all possible learner completion paths.
2. ADL Certification Test Suite Run-Time Environment Option log files.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Old Validation Method	New Validation Method
#19 Packages - Full disclosure	#19 Package-Rules Delivery Rules - D1 Rules	Army Resource Validator and ADL Certification Test Suite	ADL Certification Test Suite
#20 Packages - Consumables	#20 Package-Consumables Delivery Consumables - D2	Not Applicable	Not Applicable
NA	Delivery Documentation - D3 1. Screen Shots of Learner Completion Paths 2. Certification Test Suite log files showing Army Mandatory SCORM® data calls for content and test SCOs.	Manual Validation during Government Acceptance Review	Manual Validation during Government Acceptance Review

Table 2.3f

2.4 Evaluating the new set of Army Business Rules

This table shows whether a Business Rule results from a SCORM® requirement or an Army requirement plus identifies a benefit(s) and why a Business Rule is needed.

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Is this Business Rule a SCORM® or Army requirement?		Benefit -or- Capability most affected by compliance	Why Needed? What Capability is Supported
		SCORM	Army		
#01 No course-to-course hyperlinks	#01Hyperlink scope\ Instructional Rules - I1 Rules	No	Yes	Portability/ Reuse	Reuse
#03 Internal Context only; no sequence or hierarchical references	#03Local context	No			
#02 Content Matches TOC Title	#02Titles match Instructional Rules - I2 Titles	No	Yes	Instructional/ Learner	For Learner's benefit
#04 Must contain all critical instructional material	#04All instruction Instructional Rules – I3 Teaching	No	Yes	Portability/ Reuse	Reuse
#13 Metadata required for packages, aggregations, and content objects	#13Metadata Required Metadata Rules - M1 External	No	Yes	Reusability, Discoverability	Reuse
#14 No inline metadata	#14Metadata in files Metadata Rules - M2 Schemas	No	Yes	Usability, Best Practices	Army Regulation
#15 Schemas in Root and in folders with metadata files is required for test tools to operate properly.	#15Metadata Schema Metadata Rules - M3 Required	Yes	Yes	Support Tool compatibility , excitability	SCORM/ Army
#08 URLs must use escaped character codes	#08Escape Characters Filename Rules - F1 Characters	No	Yes	Functionality, Usability, Portability, Reuse, Best Practices	Web
#07 Illegal filename characters	#07Bad file Characters	No			

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Is this Business Rule a SCORM® or Army requirement?		Benefit -or- Capability most affected by compliance	Why Needed? What Capability is Supported
		SCORM	Army		
#11 SCORM® manifest objectives scope is local, not system (LMS)	#11Manifest Objective tag SCORM® Rules – S1 Objectives	No	Yes	Functionality, Portability/ Reuse	SCORM/ Army
#12 Assessment must post Army mandatory test item elements	#12Item Analysis SCORM® Rules – S2 Item Analysis	No	Yes	Functionality	Army Regulation
#17 SCORM® assessments must fully use an imsmanifest.xml tag for minimum mastery score	#17Manifest Score SCORM® Rules – S3 Pass	No – Best Practice	Yes	Reusability, Reducing Life Cycle Cost	Reuse
	SCORM® Rules – S4 Methods and Data for non-scoring SCOs		Yes	Functionality, Usability	Army Requirement
	SCORM® Rules – S4 Methods and Data for Scoring SCOs		Yes	Functionality, Usability	Army Requirement, Question/Item Analysis
	SCORM® Rules – S6 Error Handling for Scoring SCOs		Yes	Functionality, Usability	Army Requirement
#10 Must bookmark learner progress	#10Bookmark Code Rules – C1 Bookmarking	No	Yes	Usability	Usability, increases user friendliness
	Code Rules - C2 Answer Data Processing	No	Yes	Test Integrity	Test Integrity
#16 Graceful Exits: All Content Objects communicate termination data for all possible learner selected exit paths	#16Easy Exit Code Rules – C3 Exit	Yes	Yes	Functionality, reduces learner frustrations by reducing content malfunctions	SCORM

Old Business Rule	Old Business Rule (Short Name)/ New Business Rule	Is this Business Rule a SCORM® or Army requirement?		Benefit -or- Capability most affected by compliance	Why Needed? What Capability is Supported
		SCORM	Army		
#19 Packages - Full disclosure	#19 Package-Rules Delivery Rules - D1 Rules	No – Best Practice	Yes	Reusability, Reducing Life Cycle Cost	Reuse
#20 Packages - Consumables	#20 Package-Consumables Delivery Consumables - D2	NA	NA	Reusability, Reducing Life Cycle Cost	Reuse
	Delivery Documentation - D3 1. Screen Shots of Learner Completion Paths 2. Certification Test Suite log files showing Army Mandatory SCORM® data calls for content and test SCOs.	No	Yes	Minimize Government Review/Processing Time	Army Requirement

Table 2.4a

3. Army Instructional Business Rules

Information provided in Section 3 up to section 3.4 is informative.

Army DL is developed following the well-known ADDIE process:

- Analysis
- Design
- Development
- Implementation
- Evaluation

3.1 SCORM® 101

Developing Instructional Content following the SCORM® specification is intended to have limited repercussions in the Instructional Design of the material. Limiting repercussions is also dependent upon the SCORM® content being “in harmony” and not in conflict with the LMS capabilities and features. This section provides introductions to assist the e-learning programmer in becoming familiar with the SCORM®. This section is a very brief Army version of what is presented from the beginning up to and including chapter 6, section 1 in the Advanced Distributed Learning Initiative (ADL) document “[SCORM Users Guide for Programmers](#)”: It is highly recommended that ALL readers use the ADL document and this document together as complementary information sources.

If you are familiar with creating SCORM® packages and writing rollup rules and creating other xml tags found in a SCORM® imsmanifest.xml file, this document will make the assumption that you have a certain familiarity with the three ADL SCORM® 2004 3rd Edition Reference manuals. In that case, with a quick examination of the ADL SCORM® Users Guide for Programmers, you would be prepared to continue below with the 2.1 SCORM® Concepts and Processes section.

For readers new to SCORM®, it is extremely important that you are thoroughly familiar with the [SCORM® Users Guide for Programmers](#) chapters 1 through section 6.1 before seriously reviewing any Army Business Rules. The value in reading the ADL manual is that you are presented with fundamentals as well as multiple solutions; multiple solutions, which coincide with multiple instructional strategies, and sometimes, it is necessary to introduce multiple LMS scenarios to illustrate the influence of LMS capabilities and features.

You **always** desire to provide the instructional strategy with the properly applied solution considering the LMS capabilities and features, thereby creating the learner environment with SCORM® content “in harmony” with the LMS. After reading chapters 1 through chapter 6 section 1 of “SCORM Users Guide for Programmers” you should return to this document and continue below.

3.2 SCORM® and Instructional Design

Following SCORM® specifications and loading instructional content into an LMS will have limited repercussions in the Instructional Design of the material. This section will begin explanations that will help minimize instructional influences of SCORM® conforming content and insure that the learner experience will not be degraded simply by following specifications necessary for SCORM® compliance.

3.3 SCORM® Concepts and Processes

Key concepts and processes are introduced in this section to assist the reader in understanding this document.

3.3.1 Reusability and Independent Learning Objects

Learning content is generally produced as courses, modules, or lessons each full of resources such as animations, simulations, graphics, video, and audio. New technical specifications were developed to open these resources to discovery and reuse. These specifications allow for the separation of pieces of learning content making it available to other course developers as immediately available instructional material not needing to be developed or redeveloped.

The critical feature that enables this “portability” is creating context-neutral content by removing references to 1) external material, 2) other hierarchically related instructional material, and 3) related but external training and learning titles terms or events. The learning object becomes independent and available for total and worry free reuse by the removing all dependency on external items. Other benefits of independent, context-neutral content include:

- Use existing content to create new courses for all government
- Provide doctrinally correct instruction that can be branded by other groups
- Utilize multiple delivery channels (Internet, intranet, print and more)
- Provide for efficient and cost effective content revisions by updating content
- Improve course development time and efficiency
- Assemble new courses and other deliverables from existing content, in whole or in part

Additionally, these objects become identifiable small chunks of learning content that can be tagged with descriptive labels (metadata), which simplifies and supports discoverability by other developers.

In the SCORM® model the “small chunk” is defined as a Sharable Content Object (SCO).

3.3.2 The "Page"

The page concept has been used in computer based training for a number of years. Most authoring tools use the metaphor of a book as a basis for design layout. The learner has the look and feel of content displayed on a page with buttons to allow movement from page to page. This metaphor is repeated in both SCORM® and non-SCORM® Web-based training. Web-based

chunks are routinely delivered as HTML pages. With intra-SCO navigation, a single HTML page can appear to be more than one page to the learner and within a web-based chunk a single HTML page can appear to be more than one page to the learner.

Remember that page is just a convenient title to a presentation to a Learner; the learning resources combining to make a learning object can be media files, topics, anything that makes instructional sense.

3.3.3 SCORM® Content

SCORM® content is the files that create the learner presentations. Chapters 1, 2, and 3 in the ADL SCORM® Users Guide for Programmers cover SCORM® content including general terms, content packages, and of course, the manifest file, “imsmanifest.xml”. Also covered is how SCORM® can be used to make courses and curricula. You will not find that in the SCORM® three volume reference set since the SCORM® specifications are silent on courses, registration, and Learning management. SCORM® allows the LMS to provide all course and learner management capabilities. During the development of SCORM® specifications, LMSs were allowed to maintain functionality in the area claimed as an LMS’s unique and proprietary “Look and Feel”.

It would be good to review the ADL [“SCORM Users Guide for Programmers”](#) now. In Chapter 3 Content Packages, section 3.3 “how to implement” it is important understand the Organizations and Resources paragraphs, each with an XML sample, all near the end of section 3.3.1 “Components of a Content Package”. There is also a Best Practice that mirrors Army SCORM® Objectives Business Rule.

In section 3.3.2 the “Content Organization” paragraph should always be followed. The Army enforces what we call “full disclosure”: Full Disclosure is attained when each and every file in the package is listed in the manifest, and every file listed in the manifest is also in the package. SCORM® Content Packages should be simple and constructed with purpose; for example, a folder named “common” is a purposeful name for storing files used in multiple SCOs, and “graphics” is a purposeful folder name to hold Graphical User Interface (GUI) images. In SCORM® Content Packages, simple is better. Files should be laid out into logical folders, such as a folder for each SCO, or a folder for a file type, such as a folder named “metadata” to hold all metadata files. A SCORM® manifest should use the dependency element to identify file sets which are used in multiple SCOs.



Best Practice: No matter what tool you use to organize the files in your package, keep the file structure and organization neat and clean.

Also near the end of section 3.3.2 take notice that the Army Business Rules require more communication than the SCORM® minimum of simply calling the Initialize() and Terminate() methods.

Shareable Content Objects (SCOs) and asset files are the two basic types of Learning Content Objects used in SCORM®.

- **Asset** – Technically every file used for the learning content object is an asset. However assets can be collections of files as well. Assets are the physical files that make up the content. An asset may be a single file or a set of files. All of the graphics are assets; each image and icon that appears in a viewable screen is an asset. For example, a lesson (SCO) might contain a simulation (asset) and a glossary (asset) as well as a check on learning (asset) and an arrow icon (asset) presented for the “next” control.
- **SCOs** – Sharable Content Objects are Learning Content Learning Objects that can be launched and tracked by communicating with the learning management system. It is this communication with an LMS that makes a SCO a SCO. Without this, a SCO is just a collection of assets. SCOs should be developed to support specific instructional objectives. An entire course could be a SCO if there is no reason to break it into smaller pieces.

Learning content may be broken down into pieces to make it more reusable or to improve the learner experience.

There are limitations inherent in SCORM® that can influence design and production decisions. SCORM® specifications allow a SCO to communicate with an LMS but not with another SCO. This limitation is mandated for the SCO to be independent and reusable.

3.3.4 Army Classification of SCOs

The Army supports different classifications or purposes of SCOs relating to reusability. The same dialogue could be presented with respect to web-based non-SCORM® learning objects or chunks but in this section readers must interpret references to SCOs as references to web-based content objects in general. Independent & reusable SCOs are intended to train the task, yet be designed as context-neutral content. That is, for total and worry free reuse, SCOs must train the task without context references, such as an MOS title or skill level. Dependent SCOs have fewer restrictions and can relate to context or provide a transition between tasks. Certain rules may apply to one type of SCO and not another.

- Learners can access assets, such as remediation, help, or glossary, or references, and then return to the previous location in a SCO. Refer to the [External References](#) section (7.5). Independent SCOs are "independent," meaning those SCOs that are context-neutral learning content. Context-neutral content cannot refer to an MOS or titles of the course, module, and lesson. Army structure terms like course, phase, module, lesson, etc., should also be removed. Independent SCOs are considered portable and reusable; designed to be stored in a SCO repository and made available for other training developers to use in another course(s) with no modification required. The independent SCO may refer to non-testable reference material via a URL. Refer to the [External References](#) section (7.5).

- Dependent SCOs are not required to follow the reuse rules. Dependent SCOs can be developed to establish the context for one or more context-neutral SCOs. Course introductions, Module Introductions, and Course Overviews are examples of dependent LCOs or SCOs. Dependent SCOs could be developed to transition the learner between any other LCOs, from an introduction SCO to an independent SCO, or between two independent SCOs. Dependent SCOs are never available for total and worry free reuse; they will always require modification (repurposed) to be used in another course or even if reused in the same course. They are usually limited to a single use, but by definition, dependent SCOs are repurposed (modified) rather than reused.
- Dependent SCOs such as introduction SCOs or transitional SCOs, which do not contain learning content, may reference learning objectives, course content, etc., for the learner to understand what is to be taught. This is acceptable to the Army. The Army considers these dependent SCOs as "single use" objects and not reusable.
- Dependent SCO files will be local files contained in the content package. Dependent SCOs can contain external references accessed by URLs. Refer to the [External References](#) section (7.5).
- Learner Performance Test SCO - Learner Performance Tests are excluded from rules of independent SCOs and in fact are reusable without additional restrictions due to Army test security requirements. The Army requires Learner Performance Tests to be single SCO SCORM® Content Packages, and single SCO content packages are reusable by simple conformance to SCORM® content and packaging specifications.

The reusability of a learning resource depends on content being independent, self-contained, and context-neutral.

SCORM® recognizes, however, that some learning resources may contain internal logic to accomplish a particular learning task. Such a learning resource might branch within itself depending on user interactions. These branches are all self-contained, relevant to a stand-alone learning resource, and are not usually visible to the LMS, and not affecting total and worry free reuse. Importantly, for retaining reusability, internal branching must not reference external learning resources - they may not be available from another package or another LMS. This is an important area that content developers should pay attention to when determining what learning resources should be used and how they are to be aggregated. This restriction will sometimes result in the duplication of content (Special table of calculations needed in two objects, SCOs) and sometimes result in the creation of a reusable asset (glossary or reference reading list).

3.4 The Army Instructional Content Business Rules

The remaining material in Section 3 is normative information pertaining to the Instructional Business Rules and informational when examples and sample information is provided.



Army Business Rule I1 (Instructional Content Rules): Independent Learning Content Objects (LCOs) shall be able to stand alone as discrete blocks of learning content. The learning content of an independent LCO shall not contain references (for example, hyperlinks, narration, etc.) to another Content Object or aggregation. Independent LCO content and titles must not include sequencing numbers or other hierarchical level instruction titles or words (course, phase, module, lesson, etc.). If an independent LCO contains an ELO, "ELO" shall not be referenced within the content or title of the LCO. Independent LCO titles must not include the Military Occupational Specialty (MOS), Skill Level, Skill Qualification Identifier (SQI), or Additional Skill Identifier (ASI). Independent LCO content shall not contain the acronyms "SCO", "TLO", or "ELO"; or references to a course map or include transitional statements such as "The next lesson will explain..."

Tracked Independent Learning Content Objects (LCOs) are SCOs and Assignable Units in SCORM® and Aviation Industry Computer-Based Training Committee (AICC) Distributed Learning (DL) specifications, respectively. Untracked Independent Learning Objects may be composed of web pages or other Interactive Multimedia Instruction (IMI) presentations that stand alone as discrete blocks of learning content.

For technical details that fully present the options with respect to (WRT) the Army Business Rules being presented in this document the SCORM® 2004 3rd Edition three volume reference suite must be examined. These books consist of three reference manuals: 1) Content Aggregation Model (CAM), 2) Run-Time Environment (RTE), and 3) Sequencing and Navigation (SS&N). These three volumes are available from the ADL website as [SCORM-2004-3rd-Edition-Specification](#).

3.5 SCO Titles



Army Business Rule I2 (Instructional Titles): All Learning Content Objects (LCOs) and Sharable Content Objects (SCOs), both dependent and independent, shall display the same topic title as a Hyperlink in the Table of Contents (or Menu) and in the instructional content presented to the learner.

This section is also appropriate for non-SCORM® instruction.

The following screen capture demonstrates how a SCO title contained within the SCO displayed to the learner is the same as the title in the LMS's Table of Contents. The SCO being viewed is

titled 'Strategic Decision Making'. Likewise, the SCO title in the LMS's Table of Contents is also 'Strategic Decision Making'. Keeping the title the same in the SCO as in the table of contents makes it easier for the learner to orient him/herself within the course. It also reduces confusion and increases the ease of use.

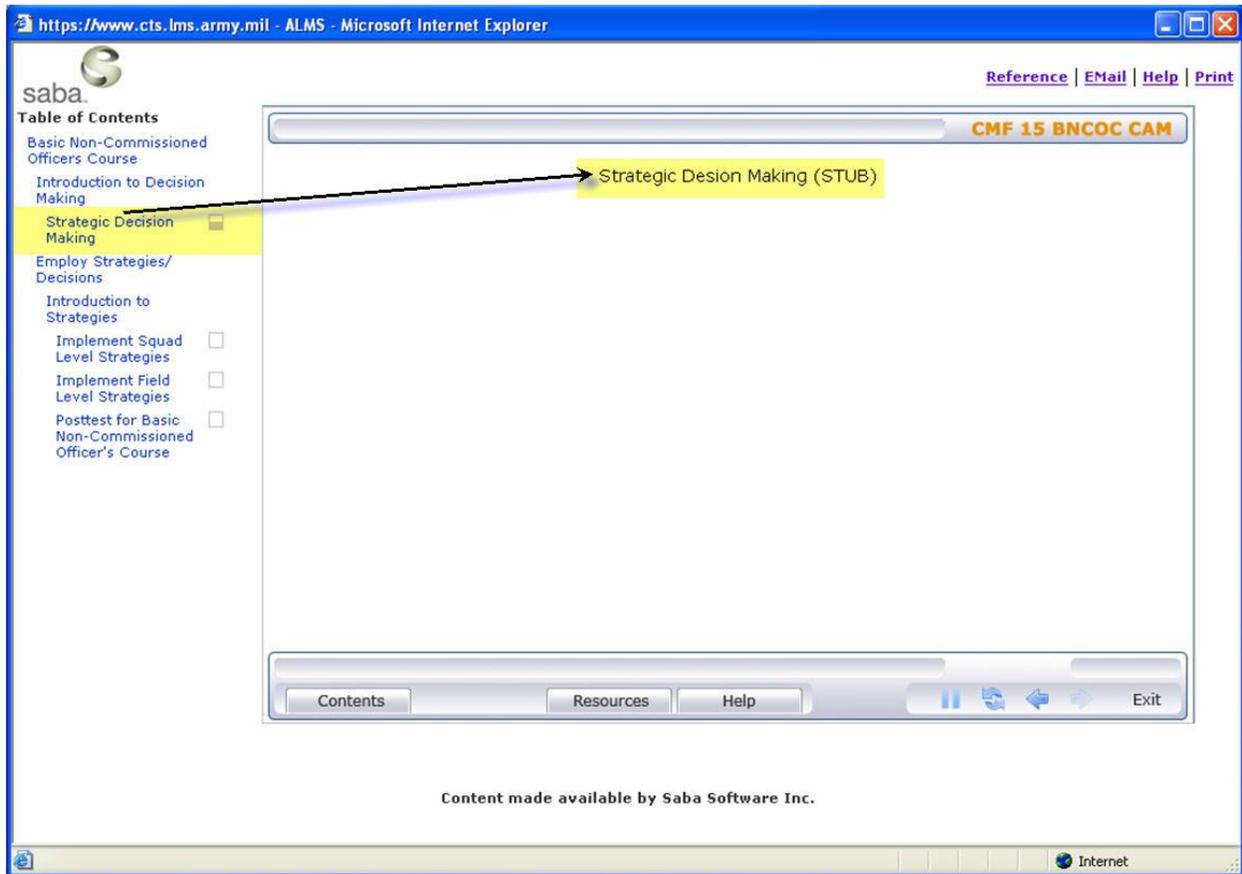


Figure 3.5a

A graphical table of contents is not needed, as titles for each SCO are eXtensible Markup Language (XML) tagged in the Organization section of the manifest file. The LMS reads the manifest file and generates the Table of Contents display. The Sequencing, Navigation, and Rollup rules also come from the Organization section.

The “Sequencing Control” navigation tags can be set to allow user choice, where the table of contents is displayed within the LMS and the learner can choose any learning activity. If user choice is not desirable, the programmer can provide instructions to the LMS to automatically launch learning activities in the proper order.

The following section has been edited to include non-SCORM® content in its presentation.

SCO and content object titles reflecting the action statement of the learning objective support reuse and independence. One title exactly describing the learning content within a SCO or learning object is best and promotes reusability. A properly titled SCO or content object will be able to be reused in a different course.

When a SCO or content object designed as an Enabling Learning Object (ELO) references its Terminal Learning Object (TLO) within its content or in its title, that SCO or content object is no longer reusable, since it is no longer independent – it is now identified (or dependent) with its TLO. A properly titled SCO or learning object allows reuse at any learning object level. SCOs and learning objects properly named and designed as ELOs may be reused as TLOs or as a learning step in another course.

SCO and content object titles should be displayed on every page of the content to keep the learner aware of where he/she is within the learning content.

A properly titled SCO or learning object does not have numerical sequencing. Reusing SCOs and learning objects that display numerical sequencing such as Chapter 1 or Topic 1 in the title will mislead and confuse the learner.

3.5.1 SCO Example

Following is an example of the XML tagging for a SCO within the manifest with a SCO defined by the 'SCORMType' attribute on the <resource> element. The 'SCORMType' must have the value of "sco". File references must be relative paths from the location of the manifest file.

```

<organizations default="TOC1">
  <organization identifier="TOC1"
    adlseq:objectivesGlobalToSystem="false">
    <item identifier="S100" identifierref="R_S100">
      <title>SCO Title goes here</title>
      <imsss:sequencing>...</imsss:sequencing>
    </item>
    ...
  </organization>
</organizations>

<resources>
  <resource identifier="R_S100" type="webcontent"
    adlcp:SCORMType="sco" href="unit1/index.html">
    <metadata>
      <adlcp:location>metadata/unit1_sco.xml</adlcp:location>
    </metadata>
    <file href="unit1/index.html"/>
    <file href="unit1/scopage2.html"/>
    <file href="unit1/scopage3.html"/>
    <file href="images/graphic1.gif"/>
  </resource>
  ...
</resources>

```

Figure 3.5.1a

3.5.2 Army Independent LCO “Essential” Instructional File Placement



Army Business Rule I3 (Instructional Learning): Instructional content files for all independent Learning Content Objects (LCOs) must be physically located within the LCO content package (within the root folder or subfolder) and not referenced by a Uniform Resource Locator (URL).

Learning content files for all Independent SCOs must be physically located within its SCORM® Content Package (within the root folder or subfolder) and not referenced by a URL.

PROGRAMMER INFO: The government may allow exceptions for non-critical material, usually for large manuals such as Soldier’s Manual of Common Tasks, Field Manual, Soldier Training Publication, and so on. All files residing external an LCO package shall be protected from removal, replacement, or updates by being stored in an Army repository or within an Army LMS. URLs for these external files shall be fully qualified.

4. Army Metadata Business Rules

In Section 4, all Army Business Rule language, specified field names and specific entry text is normative information. However information providing sample entries and instructions describing sample entry text is informational.

4.1 XML Binding for Separate Metadata Files



Army Business Rule M1 (Metadata External):

Metadata must be in separate eXtensible Markup Language (XML) files and referenced from within the manifest file using the XML binding for separate metadata file references.

The XML binding to be used within the manifest that references the relative path for the separate XML metadata files is shown in the following example.

XML binding for external metadata files contained in the SCORM® Content Package:

```
<metadata>
  <adlcp:location>[relative path from the imsmanifest.xml file to the metadata
file]</adlcp:location>
</metadata>
```

Figure 4.1a

4.2 SCORM® Metadata Schema Version

SCORM® requires that all manifest files contain the schema and schema version to which the metadata is created. This XML binding is shown in the following example:

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 3RD EDITION</schemaversion>
  </metadata>
  <organizations>
    ...
  </organizations>
</manifest>
```

Figure 4.2a

4.3 Relative Path Example

Following is an example of a relative path from the 'imsmanifest.xml' file:

If the physical file structure is:

- unitsafety folder
 - metadata folder
 - sco1_metadata.xml
 - sco2_metadata.xml
 - sco1 folder
 - sco2 folder
 - images folder
 - references folder
 - glossary folder
- imsmanifest.xml file
- imscp_v1p1.xsd
- adlseq_v1p3.xsd
- imsss_v1p0.xsd
- adlcp_v1p3.xsd
- adlnav_v1p3.xsd

If the metadata file name is sco1_metadata.xml and located in the 'metadata' folder, then, using this scenario, the <adlcp:location> tag within the manifest would be 'metadata/sco1_metadata.xml', which is the path from the location of the 'imsmanifest.xml' file to the 'sco1_metadata.xml' file. In the above file structure, any SCO can be extracted and reused in another content aggregation.

In essence, a metadata reference is contained within the manifest file and the actual metadata XML file is located within the physical files.

4.4 Content Organization Metadata Referenced on the Manifest

Content Organization Metadata is located in the <organization> section of the manifest.

Following is an example of the placement of the Content Organization metadata reference within the manifest:

```
<organization>      (opening tag for organization)
  <item>...</item>
  <item>...</item>
  <item identifier="B10">
    <item>...</item>
    <item>...</item>
    <item>...</item>
  </item>
  <metadata>      (Content Organization metadata for <organization>)
    <adlcp:location>metadata/co_unitsafety.xml</adlcp:location>
  </metadata>
</organization> (closing tag for organization)
```

Figure 4.4a

Content Organization metadata as shown here is 'co_unitsafety.xml'.

4.5 SCO Metadata Referenced on the Manifest

The SCO metadata reference is located in the <resource> section within the manifest file.

Following is an example of the placement of the SCO Metadata reference:

```
<resource identifier="A23" type="webcontent" adlcp:SCORMType="sco"
href="launchfile.htm">
  <metadata>      (SCO metadata)
    <adlcp:location>metadata/unit.xml</adlcp:location>
  </metadata>
  <file>...</file>
  <file>...</file>
</resource>
```

Figure 4.5a

4.6 Army Metadata Schema Business Rule



Army Business Rule M2 (Metadata Schema):

The metadata schemas that must reside in the root folder must also be located in the folder(s) where the metadata resides. Folders which contain metadata shall also have the metadata schemas and must be copied into this folder.

Army Business Rule 9.26-1 exists as a safeguard. Certain XML processors will try to locate the XML schema documents for the metadata by scanning the folder where the metadata resides. This action may cause some errors with XML parsing if the schema files are not present.

The schemas for all three Learning Object Metadata (LOM) Profiles can be downloaded from the Institute of Electrical and Electronics Engineers (IEEE). Link: <http://ltsc.ieee.org/xsd/lomv1.0/20040413/>.

4.7 Setting up SCORM® Metadata Schemas

The metadata schemas are required to be placed in the root folder of the content package, according to the SCORM® specification pertaining to all SCORM® schemas.

Following is a representation of how the metadata schemas are to be structured in the courseware package root folder:



Figure 4.7a

4.8 Army Metadata Requirements



Army Business Rule M3 (Metadata Required):

SCORM® content: Metadata is required for ALL Content Packages and Resource Packages, all manifest Organization level items including each SCO and all launchable assets.

Non-launchable Assets and resource aggregations must have metadata only upon the government's request.

Non-SCORM® content: Learning Content Objects that are not required to follow SCORM® communication specifications have the same metadata requirements. The government may specifically identify Learning Content Objects as exempt from Metadata requirements.

All metadata must be formatted in accordance with the instructions in section 4, Army Metadata Business Rules, and Appendix B: Metadata Tables.

The Army requires two types of metadata: Content Organization Metadata and SCO or Object Metadata.

4.9 Developing Army required SCORM® Metadata Files

Here is a macro example of how the contents of a separate metadata file named 'sco1.xml' should appear:

```
<?xml version="1.0"?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOM"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOM lomStrict.xsd">

  <general>...</general>
  <lifeCycle>...</lifeCycle>
  <metaMetadata>...</metaMetadata>
  <technical>...</technical>
  <rights>...</rights>
  <classification>...</classification>
  <classification>...</classification>
  <classification>...</classification>
  <classification>...</classification>
  <classification>...</classification>
</lom>
```

Figure 4.9a

Following is a table cross-references Army required Metadata elements with descriptions of the entry content or defined vocabulary entry for Army mandatory metadata entries.

Metadata Required	XML Metadata Tag for Programmer
Catalog Identifier	general.identifier.catalog (value = ALCMC)
Entry Identifier	general.identifier.entry (value = TBD)
Title of Learning Resource	general.title
Language of Learning Resource	general.language
Description of Learning Resource	general.description
Keywords	general.keyword
Type of Metadata	general.aggregationLevel
Version of Learning Resource	lifeCycle.version
Status of Package Submittal	lifeCycle.status (value = final)
Proponent's Role	lifeCycle.contribute.role (value = publisher)
Proponent Name, Address, E-mail	lifeCycle.contribute.entity
Date of Submittal	lifeCycle.contribute.date
Metadata Catalog Identifier	metaMetadata.identifier.catalog (value = TBD)
Metadata Entry Identifier	metaMetadata.identifier.entry (value = TBD)
Metadata Specification Used to Create this Metadata	metaMetadata.metadataSchema (value = LOMv1.0, value = SCORM_CAM_1.3, value = ADL_v1.0)
Language of Metadata file	metaMetadata.language
File Format (MIME types)	technical.format
Cost of Learning Resource	rights.cost (value = none)
Copyright and Other Restrictions	rights.copyrightAndOtherRestrictions (value = none)
MOS and Skill Level	classification (value = "discipline")
SQI	classification (value = "discipline")
ASI	classification (value = "discipline")
Task Numbers and Task Descriptions	classification (value = "educational objective")
Learning Objectives (Action, Condition, Standard)	classification (value = "educational objective")
508 Compliant	classification (value = "accessibility restrictions")
Security Level (Foreign disclosure)	classification (value = "security level")
ADL-R Requirement	Classification (value = "collection")

Figure 4.9b

4.9.1 Catalog and Entry Identifier

The catalog where this learning resource is stored is designated with the tag `general.identifier.catalog` and the entry is designated with the tag `general.identifier.entry`.

The `general.identifier` metadata element represents a mechanism for assigning a globally unique label and contains two sub-elements: `catalog` and `entry`. These two sub-elements will designate the catalog and entry identification used by the Government (ATSC) to store the courseware after delivery. The catalog element must be "ATIA" and the entry element must be "TBD".

Following is an example of designating the catalog and entry identifier:

```
<general>
...
  <identifier>
    <catalog>ATIA</catalog>
    <entry>TBD</entry>
  </identifier>
...
</general>
```

Figure 4.9.1a

4.9.2 Title of Learning Resource

The title of the learning resource is designated with the tag `general.title`.

The `general.title` metadata element describes the name given to the learning resource. For SCOs or aggregations, this value must contain the same title used in the manifest file.

Following is an example of `general.title`:

```
<general>
  <title>
    <string>Introduce Basic M60 Machine Gun Training</string>
  </title>
...
</general>
```

Figure 4.9.2a

4.9.3 Language of Learning Resource

The language (English, Spanish, German, etc.) of the learning resource is designated with the tag `general.language`.



ADL Reference: The `general.language` metadata element is required for SCOs and Content Organizations. This element must use the language code as defined by ISO 639:1988 and, if needed, the language subcode from ISO 3166-1997.

PROGRAMMER INFO: SCORM® 2004 3rd Edition defines the general.language element as being able to contain a value of "none", which indicates no lingual content.

The language of the learning resource content is specified in this element with the two-character country code.

Following are several sites that list the language codes:

- http://www.unicode.org/cldr/charts/latest/supplemental/scripts_and_languages.html
- <http://alis.isoc.org/langues/iso639.en.htm>
- <http://www.oasis-open.org/cover/iso639a.html>

Following is an example of general.language:

```
<general>
  ...
  <language>en</language>           (English)
  <language>de</language>          (German)
  ...
</general>
```

Figure 4.9.3a

4.9.4 Description of Learning Resource

The description of the learning resource is designated with the tag general.description.

The 'general.description' metadata element must contain a general description of the object for Content Organizations and SCOs. This element must exclude any reference to hierarchy (lesson, phase, module, etc.).

Following is an example of general.description:

```
For Content Organizations:
<general>
...
  <description>
    <string> Basic instruction on U.S. Army Infantry M60 Machine Gun
    laying using field expedients, range card preparation, maintenance,
    function check performance, loading, unloading, malfunction corrections
    and target engaging
    </string>
  </description>
...
</general>

For SCOs:
<general>
...
  <description>
    <string>Explanation of the field expedient method of laying an
    M60 on preselected targets using the notched stake/tree crotch method
    </string>
  </description>
...
</general>
```

Figure 4.9.4a

4.9.5 Keywords

Keywords are designated with multiple tags of 'general.keyword'.

The 'general.keyword' metadata element contains keywords or phrases describing this learning resource. One or more keyword elements are derived from description. Enter words and phrases that accurately and precisely define the object.

It is up to the content developer to determine what is represented as this string value.

Following is an example of 'general.keyword':

```
<general>
...
  <keyword><string>Infantry</string></keyword>
  <keyword><string>M60 laying using field expedients</string></keyword>
  <keyword><string>Notched Stakes</string></keyword>
...
</general>
```

Figure 4.9.5a

4.9.6 Type of Metadata

Since there are five types of metadata, the type of metadata must be designated with the tag 'general.aggregationLevel' to determine proper testing requirements. The value is converted to a numeric value being "2" for a SCO or "3" for a collection of SCOs.

The general.aggregationLevel metadata element must designate which of the two types of resources the metadata is describing that the Army requires: SCO metadata or Content Organization metadata.



ADL Reference: The SCORM® specification states that this data element container describes the "functional granularity of the learning resource." The vocabularies defined for this element are restricted vocabularies as follows: "1" the smallest level of aggregation, for example, raw media data or fragments; "2" a collection of level 1 learning objects, for example, a lesson; "3" a collection of level 2 learning objects, for example, a course; and "4" the largest level of granularity, for example, a set of courses that lead to a certificate.

The Army has interpreted SCORM®'s four granularities of learning resources to mean the following (SCORM® Content Packages are at the course level and below for all Army DL):

- "1" indicates that an Asset is being described (Metadata is not required for Assets)
- "2" indicates that a SCO is being described
- "3" indicates that a collection of SCOs or an Aggregation is being described (Cluster)
- "4" indicates that a Content Organization is being described (Package or Course)

Following is an example of the metadata tag for SCO level metadata:

```
<general>
...
  <aggregationLevel>
    <source>LOMv1.0</source>
    <value>2</value>
  </aggregationLevel>
...
</general>
```

Figure 4.9.6a

4.9.7 Version of Learning Resource

The version number of the learning resource is designated with the tag lifeCycle.version.

The lifeCycle.version metadata element describes the edition of the learning resource. Initial version will be 1.0. Corrections will change minor version number; enhancements/updates will

change major version number. This element upon initial final delivery to the Army must contain the value "1.0".

Example: 1.0 for the initial final delivery:

```
<lifeCycle>
  ...
  <version>
    <string>1.0</string>
  </version>
  ...
</lifeCycle>
```

Figure 4.9.7a

4.9.8 Status of Package Submittal

The submittal status of "final" for the learning resource is designated with the tag `lifeCycle.status` tag and shown in the following example:

```
<lifeCycle>
  ...
  <status>
    <source>LOMv1.0</source>
    <value>final</value>
  </status>
  ...
</lifeCycle>
```

Figure 4.9.8a

4.9.9 Proponent's Role

The role of the proponent as contributor is designated as "publisher" with the 'lifeCycle.contribute.role' tag and shown in the following example:

```
<lifeCycle>
  ...
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value> publisher</value>
    </role>
  </contribute>
  ...
</lifeCycle>
```

Figure 4.9.9a

4.9.10 Proponent's Name and Address

The 'lifeCycle.contribute.entity' metadata element identifies the organization (proponent) contributing to this resource according to the vCard specification located at <http://www.imc.org/pdi/pdiproddev.html>, the "internet mail consortium site. This value must contain the full name, address, school code, and e-mail of the proponent institution according to the vCard specification and designated with the 'lifeCycle.contribute.entity' tag.

Example of the vCard format is as follows:

```
<lifeCycle>
  ...
  <contribute>
    <entity>BEGIN:VCARD
VERSION:2.1N:U.S. Army Infantry School
ORG:U.S. Army;Army Infantry School;Fort Benning
NOTE:071
ADR;DOM;WORK:Suite 650;6751 Constitution Loop;Fort Benning;GA;31905-
4502;U.S.
EMAIL;INTERNET:soldierinfo@benning.army.mil END:VCARD
    </entity>
  </contribute>
  ...
</lifeCycle>
```

Figure 4.9.10a

The 'NOTE' field has been designated as the method for identifying the school code. A listing of all school codes are contained in TRADOC Regulation 350-70, Appendix C, section C-2.

4.9.11 Date of Submittal

The 'lifeCycle.contribute.date' metadata container element describes the date of the submission. This element must contain the date that the object was approved final using the SCORM® designated format of YYYY-MM-DD.

The date of submittal is designated with the 'lifeCycle.contribute.date' tag and shown in the following example:

```
<lifeCycle>
  ...
  <contribute>
    <date>
      <datetime>2004-07-10</datetime>
    </date>
  </contribute>
</lifeCycle>
```

Figure 4.9.11a

4.10 Meta-Metadata Catalog and Entry Identifier

The 'metaMetadata.identifier' container element represents a mechanism for assigning a globally unique label that identifies the metadata record that describes the SCORM® Content Model Component. This container element contains two sub-elements of <catalog> and <entry>.

<Catalog> represents the name or designator of the identification or cataloging scheme for the entry. <Entry> represents the value of the identifier within the identification or cataloging scheme that designates or identifies the metadata.

The Meta-Metadata catalog identifier is designated with the tag 'metaMetadata.identifier.catalog'. The Meta-Metadata entry identifier is designated with the tag 'metaMetadata.identifier.entry'.

The catalog element requires the value of "TBD" and the entry element requires the value of "TBD" as shown in the following example:

```
<metaMetadata>
  <identifier>
    <catalog>TBD</catalog>
    <entry>TBD</entry>
  </identifier>
  ...
</metaMetadata>
```

Figure 4.10a

4.10.1 Meta-Metadata Schema

The Meta-Metadata schema is designated with the element 'metaMetadata.metadataSchema'.

The 'metaMetadata.metadataSchema' metadata element identifies the name and version of the authoritative specification used to create this metadata instance. The Army requires three elements: 1) an element with a value of "LOMv1.0", 2) an element with the value of "SCORM_CAM_v1.3", and 3) an element with the value of "ADLv1.0".

PROGRAMMER INFO: Requiring these three values will ensure that Army metadata will be valid under both SCORM® 2004 2nd Edition and SCORM® 2004 3rd Edition.

```
<metaMetadata>
  ...
  <metadataSchema>LOMv1.0</metadataSchema>
  <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
  <metadataSchema>ADLv1.0</metadataSchema>
</metaMetadata>
```

Figure 4.10.1a

The metadata instances for all of the SCORM® metadata must conform to both the conformance requirements of the LOM and the SCORM® CAM Reference Manual.

4.10.2 Language of the Metadata File

The language (English, Spanish, German, etc.) of metadata file is designated as 'metaMetadata.language' to indicate the language of all <string> elements within the metadata file. If this value is provided, then it is not necessary to indicate a language attribute for the <string> elements.

The 'metaMetadata.language' element identifies the language contained within the metadata tags of the XML file. Use the same language codes as the general.language element.

```
<metaMetadata>
  ...
  <language>en</language>
</metaMetadata>
```

Figure 4.10.2a

4.10.3 File Formats

The 'technical.format' metadata element identifies all the technical data types of this resource by using the Multipurpose Internet Mail Extensions (MIME). This element must contain all MIME types that are used in the learning resource being described.

MIME types are used to identify the software needed to access the resource. Browsers use MIMEs in the same way they use legends. If a browser received some content with a unique file extension, it looks in its list of MIME types to help it identify the content. The syntax of MIME types is as follows: type + "/" + subtype. These types are then associated with a file extension. For more details on MIME types, go to:

- MHONARC.org – the MIME information page of a Perl mail-to-HTML converter program
- IANA.org - official list maintained by the Internet Assigned Numbers Authority (IANA).

Following is an example of 'technical.format':

```
<technical>
...
<format>text/html</format>
<format>text/plain</format>
<format>image/gif</format>
<format>image/jpeg</format>
<format>application/x-shockwave-flash</format>
...
</technical>
```

Figure 4.10.3a

4.10.4 Cost of Learning Resource

The 'rights.cost' metadata container element describes whether use of the resource requires payment. This element must have the <source> element equal to "LOMv1.0" and the <value> element equal to "no".

Following is an example of rights.cost:

```
<rights>
  <cost>
    <source>>LOMv1.0</source>
    <value>no</value>
  </cost>
...
</rights>
```

Figure 4.10.4a

4.10.5 Copyright and Other Restrictions

The 'rights.copyrightAndOtherRestrictions' metadata element indicates whether copyright or other restrictions apply to the use of this resource. This value must be the <source> element equal to "LOMv1.0" and the <value> element equal to "no".

Following is an example of 'rights.copyrightAndOtherRestrictions':

```
<rights>
...
  <copyrightAndOtherRestrictions>
    <source>>LOMv1.0</source>
    <value>>no</value>
  </copyrightAndOtherRestrictions>
</rights>
```

Figure 4.10.5a

4.10.6 Classification

The classification container element describes where this resource is placed within a particular classification system. To define multiple classifications, there may be multiple instances of this category. This element is required for Content Organization and SCO metadata. This classification container element must have three (3) sub-elements per classification instance, namely, <purpose>, <description>, and <keyword>. Include one instance each for MOS & Skill Level, Tasks, Learning Objectives, Accessibility Restrictions, Security Level, and Collection. If courseware is classified by a SQI or ASI or other classification, then one instance each must also be provided.

The 'classification.purpose' best practice vocabulary relates to Army terms as follows:

Best Practice Vocabulary	In Army Terms
Discipline	MOS & Skill Level
Discipline	SQI
Discipline	ASI
Educational objective	Task Numbers and Task Descriptions
Educational objective	Learning Objectives (Action, Condition and Standard)
accessibility restrictions	Section 508 compliant or not
security level	Foreign Disclosure
collection	DOD

Figure 4.10.6a

For an MOS and Skill Level, enter the MOS and Skill Level and textual description of the MOS for which the object was designed. Following is an example of this classification instance:

```
<classification>
  <purpose>
    <source>>LOMv1.0</source>
    <value>discipline</value>
  </purpose>
  <description><string language="en-US">11C2 Indirect Fire Infantryman</string>
  </description>
  <keyword><string language="en-US">Indirect Fire Infantryman</string></keyword>
</classification>
```

Figure 4.10.6b

For an SQI, enter the SQI and textual description for which the object was designed. Following is an example of this classification instance:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>discipline</value>
  </purpose>
  <description><string language="en-US">E Mountaineer</string>
</description>
  <keyword><string language="en-US">Mountaineer</string></keyword>
</classification>
```

Figure 4.10.6c

For an ASI, enter the ASI and textual description for which the object was designed. Following is an example of this classification instance:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>discipline</value>
  </purpose>
  <description><string language="en-US">Q6 Long Range Surveillance
Leader<string>
</description>
  <keyword><string language="en-US">Long Range Surveillance
Leader</string></keyword>
</classification>
```

Figure 4.10.6d

For Tasks, enter the list of task numbers and titles of the critical tasks for which this SCO/Organization provides training or support. Following is an example of this classification instance:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>educational objective</value>
  </purpose>
  <description><string language="en-US">071-312-3003 Lay An M60 Machine gun
Using Field Expedients; 071-312-3007 Prepare A Range Card For An M60 Machine
gun; 071-312-3025 Main An M60 Machine Gun<string>
</description>
  <keyword><string language="en-US">Range Card </string></keyword>
</classification>
```

Figure 4.10.6e

For Learning Objectives, enter the Action, Condition, and Standard for which this SCO/Organization provides training or support. Following is an example of this classification instance:

```
<classification>
  <purpose>
    <source><langstring xml:lang="x-none">LOMv1.0</langstring></source>
    <value><langstring xml:lang="x-none">educational
objective</langstring></value>
  </purpose>
  <description><string language="en-US">Action: Lay An M60 Machine gun Using
Field Expedients; Condition: Given Interactive Multimedia Instruction;
Standard: The Standards are met when the learner has completed the IMI lesson
and achieved a passing score on a separately administered test.<string>
  </description>
  <keyword><string language="en-US">M60 Machine gun using Field
Expedients</string></keyword>
</classification>
```

Figure 4.10.6f

For accessibility, enter whether or not this object is Section 508 compliant. The values for <description> are "508 Compliant" or "Not 508 Compliant" and the values for <keyword> are "508", "PL508", or "accessibility restrictions". Following is an example of a classification instance:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>accessibility restrictions</value>
  </purpose>
  <description><string language="en-US">Not 508 Compliant<string>
  </description>
  <keyword><string language="en-US">Not 508</string></keyword>
</classification>
```

Figure 4.10.6g

For Foreign Disclosure, the following is an example of a classification instance:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>security level</value>
  </purpose>
  <description><string language="en-US">FD1<string>
  </description>
  <keyword><string language="en-US">FD1</string></keyword>
</classification>
```

Figure 4.10.6h

For the ADL-R, the following is an example of a minimum classification instance:

```
<classification>
  <purpose>
    <source>ADL-Rv1.0</source>
    <value>collection</value>
  </purpose>
  <taxonPath>
    <source>
      <string>ADL/DOD Content Category Taxonomy</string>
    </source>
    <taxon>
      <entry>
        <string>DOD</string>
      </entry>
    </taxon>
  </taxonPath>
</classification>
```

Figure 4.10.6i

5. Army Filename Characters and File Reference Business Rule

5.1 File Naming Convention



Army Business Rule F1 (Filename Characters): The Army file and folder naming requirements follow the Network Working Group Request for Comments 3986, commonly referred to as **RFC 3986** – Uniform Resource Identifier (URI): Generic Syntax.

Forbidden Characters:

The two delimiter character sets, Generic and Subcomponent, make up the first group of characters the Army does not allow in file or folder names. These characters are presented below using a vertical bar “|” delimiter:

“.” | “/” | “?” | “#” | “[“ | “]” | “@” | “!” | “\$” | “(“ | “)” | “*” | “+” | “,” | “;” | “=” | ’ (right single quote)

Additional forbidden characters the Army forbids to be in file or folder names are characters found or reported to cause problems in a UNIX® Operating System Environment. These characters are presented below using a comma “,” delimiter:

|, <, >, \, ”, ` , ~, {, }, &, ^, <space>, <tab>

External resource names, which contain forbidden characters, shall be referenced in Army DL content with a URL that percent encodes each forbidden character.

Allowable Characters:

Allowable characters in Army DL file and folder names are part of the RFC 3986 unreserved character set: alpha (lowercase only) | digit | “-“ | “_” | “.”

The full list characters the Army allows in file and folder names:

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0,1,2,3,4,5,6,7,8,9, -, _, . (period)

Additional Army File and Folder Naming Guidance:

- A single period shall be used to separate the file name from its extension. Folder names shall not contain a period.
- Long file names are allowed but must conform to ISO 9660 with Microsoft Joliet extension and the Army's file naming scheme. Joliet allows filenames up to 64 characters in length. Joliet allows spaces but the Army's naming scheme does not.

This Business Rule reduces many of the human errors associated with the consistent naming of URLs, files, and folders names that have upper and lower case letters.

Examples of valid filenames:

unitsafety-sco-a328.html

p1180_launch.html

ca217-metadata.xml

sco_21.xml

m60-graphic.gif

Figure 5.1a

Handling of Filenames in GFI

Example for file naming within a resource content package: Contractor receives, as GFI, a file named 'AR 5-23 Unit Safety.pdf'. This file mixes use of upper and lower case characters and uses the restricted 'space' character. Before using the file in the courseware, it is renamed as 'ar_5-23_unit_safety.pdf', replacing the space with an underscore and uppercase letters with lowercase.

6. Army Scripting and Coding Business Rules

6.1 Bookmarking



Army Business Rule C1 (Code Bookmarking): Sharable Content Objects (SCOs) and other learner progress tracking Learning Content Objects (LCOs) shall provide each learner the ability to bookmark the learner's progress whenever the learner exits prematurely. SCOs and LCOs must allow the learner the option of resuming progress in that SCO or LCO from the bookmarked location posted during the previous learning session. Standard exceptions to the Bookmarking requirement exist, such as credit producing assessments, short learning objects with a single entry point, or other short learning objects such as summaries and introductions. Proponents may designate specific Bookmarking processing requirements, and proponents may also designate any SCO or LCO exempt from the Bookmarking requirement.

Bookmarking is a process where the learner or the system suspends learning for a certain period of time, and upon re-entry, the learner may return to the point within the learning content before the exit, or his last "locations". Learner "location" is not necessarily on page or the exact exit point. Many times the bookmarking is determined by a presentation or activity and bookmarking progress is from activity begin point to the next activity begin point. The normal bookmarking action is to provide the learner a choice to return to the last location or return to the SCO or Menu item first page or beginning. Bookmarking does not bookmark which SCO, or subpackage object or item the learner last accessed, Bookmarking tracks the last location accessed by the learner with each SCO, or Table of Contents (menu) item or topic.

6.1.1 Government Validation of the Bookmarking Business Rule

The Army has an automated conformance test for Army mandatory SCORM® processing to ensure each instructional content SCO has at least one SetValue command for the data element "cmi.location". This effort is performed by the Army Multi-log Parser when reading the AFLCTS Run-Time Environment Option's Log File.

6.1.2 Bookmarking Sample JavaScript

The data model element most commonly used for bookmarking is "cmi.location". This value can be stored when the learner suspends the SCO and then retrieved when the learner re-accesses the SCO.

Following are sample bookmarking script.

Retrieving:

```
// retrieving the bookmark from the LMS  
var retVal = doGetValue("cmi.location");
```

Figure 6.1.2a

Storing (pseudo code):

```
// storing the bookmark in the LMS  
doSetValue("cmi.location", "[page the learner is viewing]");
```

Figure 6.1.2b

Storing:

```
doSetValue("cmi.location", window.location);
```

Figure 6.1.2c

Upon entry into a SCO the user can be prompted when cmi.entry is “resume” and cmi.location is not empty.

6.2 Question and Answer Processing



Army Business Rule C2 (Answer Data Processing): Assessment content packages must not contain question and answer data in a format clearly recognizable by the learner. Question and answer data must be altered, or obfuscated, into a format not easily identifiable and readable by the learner:

- When the assessment process includes compiling or publishing, question and answer data must be embedded within a compiled or published file; such as a Flash '.swf' file.
- When question and answer data is in a text data source such as a JavaScript file or xml file, the text files must be obfuscated using a binary to text encoding method.

This Business Rule is all about supporting the integrity of test questions and making it difficult to access assessment compromising material. It is not all about exotic and elaborate encoding techniques. Just a simple base 16 or character translation algorithm binary to text encoding process will make the source material sufficiently obscured from simple examination.

The details for assessment logic and packaging decisions must be presented in the IMDP. The Assessment section of the IMDP must also provide details of the assessment implementation methods designed to satisfy this business rule. All necessary information for the government or

another party examining the IMDP to locate and identify the executing files in that assessment package which are used to comply with Army Business Rule C2 (Answer Data Processing).

Flash provides simple methods to alter input from an easily examined and understood SCORM® content package text file to something more difficult to examine and understand, by making the text file part of an encoded/compiled "executing" .swf file.

Flash use cases:

#1) A flash based assessment routine reads an external ActionScript (AS) file for questions, and assume the AS file "evaluate-midrange_smoke.as" is a question source file for a lesson assessment. "evaluate-midrange_smoke.as" must not be in the SCORM® Package; to satisfy Army Business Rule C2 Answer Data Processing, the file "evaluate-midrange_smoke.as" must be included during the publishing step.

#2) A flash based assessment routine reads an XML marked up file for questions, and assume file "evaluate-midrange_smoke.xml" is the pool question source file for a lesson assessment. "evaluate-midrange_smoke.xml" must not be in the SCORM® Package. This Army Business Rule requires "evaluate-midrange_smoke.xml" be included during the publishing step so that it becomes part of the .swf file.

6.3 Communication with the LMS (Run-Time Environment) Examples



ADL Reference: Every SCO is required, per SCORM® specifications, to adhere to the SCORM® Run-Time Environment. This implies that it must have a means to locate an LMS provided API Instance and must invoke the minimum API methods 'Initialize("")' and 'Terminate("")'. The SCO locates the API Instance according to the valid locations that an LMS is permitted to place its API Instance. All communication between the LMS and the SCO is initiated by the SCO.

PROGRAMMER INFO: All of the examples in this section use the JavaScript scripting language and the functionality is obtained by using JavaScript through the Run-Time Environment or the browser.

Differences between SCORM® v1.2 and SCORM® 2004 API function calls:

SCORM® v1.2	SCORM® 2004
<code>LMSInitialize("");</code>	<code>Initialize("")</code>
<code>LMSGetValue(data model, value);</code>	<code>GetValue(data model, value);</code>
<code>LMSSetValue(data model, value);</code>	<code>SetValue(data model, value);</code>
<code>LMSCommit("");</code>	<code>Commit("");</code>
<code>LMSFinish("");</code>	<code>Terminate("");</code>

Figure 6.3a

The following API General Application Rules shall be followed in order to achieve interoperability:

- The function names are all case-sensitive, and must always be expressed exactly as shown and described in the SCORM® specification.
- The function parameters or arguments are case-sensitive. All SCORM® supported data model (SCORM® Run-Time Environment Data Model and SCORM® Navigation Data Model) parameters shall be represented in lower case.
- All parameters passed between a SCO and the API Instance are treated as ECMAScript strings and shall be compatible with the data types and formats described by the data models that use the API for communication.



Best Practice: A JavaScript API Wrapper file is provided by ADL as an example of the SCORM® communication with the LMS. This file incorporates state management diagnostic functions to handle errors. Users may obtain a new API wrapper file (as of October 2011) ADL's Web site.

All of the programming code examples assume use of the API Wrapper file developed by ADL. Programming subroutines may begin with a similar name but have different parameters. It is the developer's responsibility to make sure to use the API Wrapper function calls if using the API Wrapper file or, if not, use the exact SCORM® API function calls as designated in the SCORM® specification. If not using the API Wrapper file, the developer would replace each function name listed in the following figure with the actual SCORM® specification API function call:

Using actual SCORM® API function calls without the API Wrapper file	With the API Wrapper file used in Army examples
Initialize("")	doInitialize()
GetValue("")	doGetValue()
SetValue("")	doSetValue()
Commit("")	doCommit()
Terminate("")	doTerminate()

Figure 6.3b

For example, the API Wrapper function call is 'doInitialize()' within this document, and the SCORM® specification API function call is 'Initialize()'.' There could be different versions of a Wrapper file with different function names.

6.3.1 Begin Communication



ADL Reference: 'Initialize' is the mandatory SCORM® API function call for all SCOs, which indicates that the SCO is ready to begin communication with the LMS.

```
doInitialize();
```

Figure 6.3.1a

PROGRAMMER INFO: All examples use the API Wrapper file function names.



Best Practice: Initialize should not be called from the 'onLoad' event on the first page of the SCO. Errors will result when clicking the 'Back' button on the second page.

To avoid this error, a 'SCO Start' (redirect) page has been suggested that separately calls 'Initialize'. This page is placed as the launch page of the SCO. It will call 'Initialize' and then immediately will redirect the learner to the first 'real' page of the SCO. All that is required is changing the href of the anchor tag on the 'SCO Start' page to the redirected real first page of SCO. If this method is used, 'Initialize' should only be called on the 'SCO Start' page.

Sample code for the 'SCO Start' page ('scostart.html') is as follows:

```
<html>
<head>
<script type="text/javascript" src="APIWrapper.js"></script>
</head>
<body>
Initializing Lesson...<br>
Please wait
<script type="text/javascript">
doInitialize();
<!-- (This href would be real first page) -->
document.location.href="unit/index.html";
</script>
</body>
</head>
```

Figure 6.3.1b

6.3.2 End Communication



ADL Reference: 'Terminate' is mandatory for all SCOs. 'Terminate' indicates that the SCO no longer needs to communicate with the LMS. Upon receiving a 'Terminate' request from a SCO, the LMS shall set the `cmi.exit` to "" (empty characterstring) or resume and exit the user from the package.

```
doTerminate();
```

Figure 6.3.2a

6.3.3 Army Business Rule C3 Code Exit



Army Business Rule C3 (Code Exit): Regardless of learner SCO exit method, learner progress shall be recorded consistently and the SCO must terminate communication with the LMS (for example, a SCO navigation request, LMS navigation button, closing the browser window, pushing an 'Exit' button on the SCO page, etc.).

End communication is a specific command given to end communication with the Learning Management System (LMS). It is the responsibility of the SCO to terminate communication with the LMS, which signals the LMS to store all buffered data and cease communication with the SCO. It is also the responsibility of the SCO to close windows that the SCO opened and ensure an orderly exit process is performed.

PROGRAMMER INFO: Linking to an external reference is not the same as leaving a SCO.

6.3.4 Examples of Properly Coding SCO Exits

In SCORM® 2004, a SCO may be removed from a learner by either a SCO's internal navigation buttons, a window event (i.e. closing the browser window), or through a loss of connection with the LMS. In order to ensure that 'Terminate' is called in a majority of these cases, SCO developers should design their SCOs to 'Terminate' on a page event (that is, `onBeforeUnload`). In order to avoid the error that will occur if the SCO calls 'Terminate' more than once, SCOs should be designed to track their current status. This can be implemented several ways; the frameset approach will be presented here.

Frameset Solution:

Following is the solution for a frameset, which places the exit code on the parent frame instead of the frame containing the learning content. This means that only when the page is unloaded will the events be executed. When navigating through the SCO's pages, the SCO's 'isTerminated' variable keeps track of the SCO's current status. Any time the SCO is removed, either by the learner, the LMS, or a browser window closing event, the SCO will attempt to terminate the session.

Sometimes it is necessary to call 'Terminate' from within the SCO in order to initiate a navigation request. This is often the case when SCOs implement their own navigation interface. In these cases, the SCO must call 'Terminate' first, and then the LMS will unload the SCO and execute the desired navigation event. If there is no way to detect the current status of a SCO, 'Terminate' could be called twice. This could have unanticipated results. It is therefore very important for a SCO to be able to track its termination status. If the SCO has already been terminated, then 'Terminate' should not be called again.

Using the above frameset scenario, an example of a frameset page ('frame.html') follows:

```
<html lang="en-US">
<head>
  <title>GUI Template</title>
  <script type="text/javascript" src="SCO_bestprac.js"> </script>
  <noscript>Sorry, your browser does not support JavaScript. For
    this training to run correctly JavaScript must be
    enabled.
  </noscript>
  <script type="text/javascript" src="APIWrapper.js"> </script>
  <noscript>Sorry, your browser does not support JavaScript. For
    this training to run correctly JavaScript must be
    enabled.
  </noscript>
  <script type="text/javascript">

    var isTerminated = false;

  </script>
</head>
<frameset rows="*,0px" border="0" onbeforeunload="endSCO('suspend')" >
  <frame name="content" src="scostart.html" scrolling="no"
    frameborder="0" title="Content frame, used to display
    training content." />
  <frame name="storage" src="storage.html" noresize="noresize"
    frameborder="0" title="Storage frame, used to store training
    related information." />
  <noframes>
    Sorry, your browser does not support frames. For this
    training to run correctly Frames must be enabled.
  </noframes>
</frameset>
</html>
```

Figure 6.3.4a

Notice that a variable 'isTerminated' is initialized on the parent frame. This will be used to track the termination status for the SCO.

The courseware content page ('scostart.html') is shown in the following example:

```
<html lang="en-US">
<head>
  <title>SCO Start</title>
  <script type="text/javascript" src="APIWrapper.js"> </script>
  <noscript>Sorry, your browser does not support JavaScript. For
    this training to run correctly JavaScript must be
    enabled.
  </noscript>
  <script type="text/javascript" src="SCO_bestprac.js"></script>
  <noscript>Sorry, your browser does not support JavaScript. For
    this training to run correctly JavaScript must be
    enabled.
  </noscript>
</head>

<body leftmargin="0" topmargin="0" rightmargin="0" bottommargin="0"
  marginwidth="0" onLoad=" beginSCO( 'index.html' ); ">

  <p align="center">
    
  </p>
  <h1 align="center">Initializing Training</h1>
  <p align="center">Loading...</p>

</body>
</html>
```

Figure 6.3.4b

Notice that this content page will call the JavaScript function 'beginSCO()'. Excerpts from this function are seen in the example that follows. It is important to also note that 'beginSCO()' takes in a parameter 'defaultstartpage' as a relative file location. This parameter will be loaded into the parent frameset as the content page. This is insignificant for this example, but does provide another example of developing SCOs in a modular fashion.

Excerpts from the JavaScript file ('SCO_bestprac.js') that defines how this SCO will interface with the APIWrapper is shown in the following example:

```
// =====  
// beginSCO() will start up a new sco. Generally called in the  
// onLoad of  
// body tag of the first page of a SCO  
function beginSCO( defaultStartPage ) {  
  
    doInitialize();  
    isTerminated = false;  
  
    // More business logic that gets executed on initialization of SCO  
    // ...  
  
    // Move the page!!  
    if ( defaultStartPage ) {  
        document.location = defaultStartPage;  
    }  
}  
  
// =====  
  
// End the SCO. Generally gets called when the user leaves  
// the sco in some fashion.  
function endSCO( exitStatus ) {  
  
    // Check to see if the SCO is already Terminated  
    if( isTerminated ){ return;}  
  
    // More business logic that gets executed when a SCO ends,  
    // such as setting session time, exit status, etc...  
  
    parent.isTerminated = true; //SET THIS TO ENSURE ONLY 1  
    TERMINATE  
    //isTerminated = parent.isTerminated;  
  
    doTerminate();  
}
```

Figure 6.3.4c

Notice that the 'beginSCO()' function will initialize the 'isTerminated' value to "false". The 'endSCO()' function will first check this variable to ensure that the SCO hasn't already been terminated. It will then continue to wrap up all necessary communications with the LMS.

Once the 'endSCO()' function has completed all necessary communications with the LMS, it sets the 'isTerminated' variable to "true" in the parent frame before it calls the 'doTerminate()' function from the 'APIWrapper.js' file. Now, if the 'endSCO()' function is called again, that is, the page unloads, the 'isTerminated' variable will equate to 'true', and the 'endSCO()' function will close without setting 'Terminate' twice.

6.3.5 Exit Status Example

The Army Content is required to disable LMS user controls which could allow the Learner to close a SCO from a control outside the content display window. Therefore, the Army recommends an empty string ("") to indicate a normal exit state. Using "log-out" is not recommended except in the context of completing an examination which indicates to the learner the completion of a larger event than one single SCO.

```
doSetValue( "cmi.exit", "" );
```

Figure 6.3.5a

7. Army SCORM® Business Rules

The Business Rule language in this section is normative as well as directions specific to commands and data elements necessary to implement the Business Rules. Other guidance, Best Practices, and examples in this section is informational.

7.1 Global Objectives



Army Business Rule S1 (SCORM® Objectives): The scope of all manifest global objectives shall be the manifest itself. Manifest global objectives shall be shared within a single SCORM® imsmanifest.xml file. SCORM® sets the default scope for global objectives to "true". This rule requires that the organization attribute objectivesGlobalToSystem be set to "false" in all DL content submitted to the Army. A valid organization entry is "<organization identifier="37W40 examinations" structure="hierarchical" adlseq:objectivesGlobalToSystem="false">".

The SCORM® specifications have the default manifest objectives mapped to the LMS system level. This makes a learner's result from one SCORM® Content Package possibly influence the settings for that learner in another SCORM® Content Package in the LMS. This is not desirable in the Army environment and the alternative setting is the focus of the Army SCORM® Business Rule S1 (SCORM® Objectives). At this setting the success of a learner in a SCORM® Content Package is only significant for that package. This business rule means that Army manifest objectives are for objective progress information related to one single manifest. Furthermore, the manifest objective progress information is available only in that single package and is not available to any other packages.

SCOs can indirectly access data from other SCOs with the implementation of global objectives. For example, a pretest SCO can write a value such as "passed" to a global objective. Then, other SCOs, as they are launched, can read this value and apply sequencing rules based on this value. If the objective indicates "passed", then the LMS could automatically launch the next lesson pretest or target a specific SCO for launching.

An example of the organization tag with the proper adlseq:objectivesGlobalToSystem attribute setting:

```
<organization identifier="MapReading" structure="hierarchical" adlseq:objectivesGlobalToSystem="false">
```

Figure 7.1a

Following is an example of sharing data from a pretest:

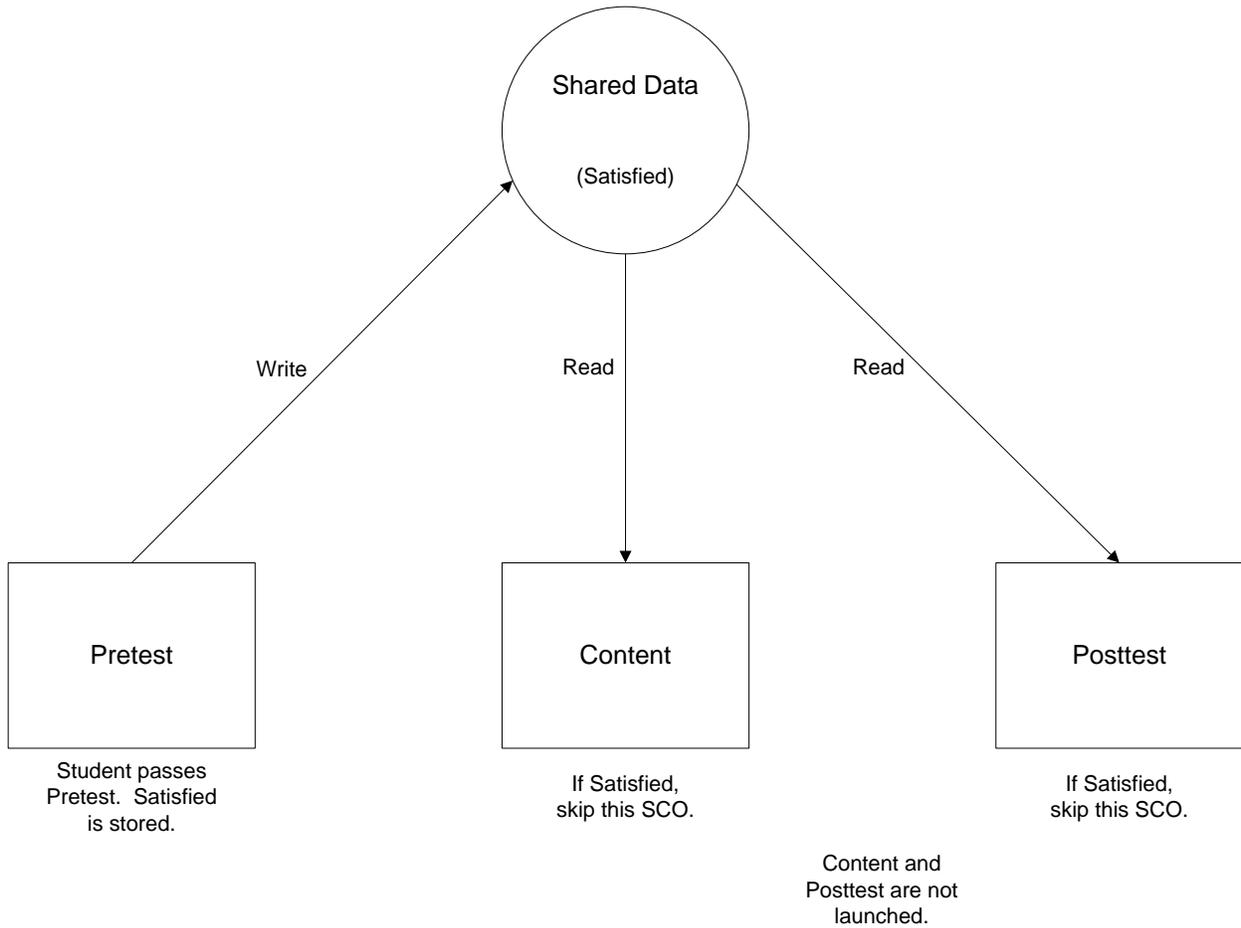
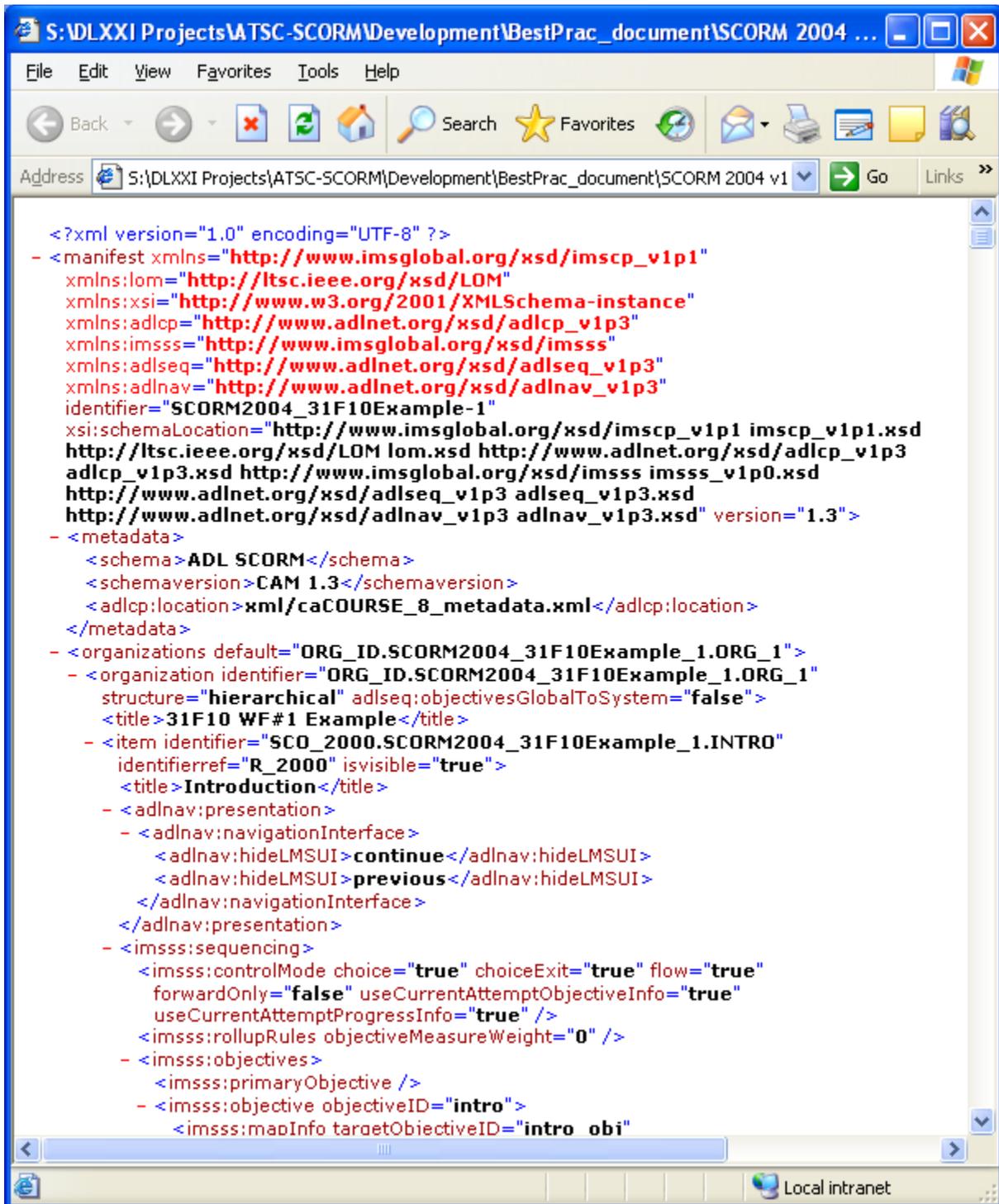


Figure 7.1b

7.2 Screenshot of Manifest File

The <organizations> section describes one or more content aggregations represented by the <organization> tag. Each <organization> tag specifies a distinct content structure such as a table of contents.

The XML tagging within the manifest file is partially shown in the following example:



The image shows a web browser window with the address bar displaying the file path: S:\DLXXI Projects\ATSC-SCORM\Development\BestPrac_document\SCORM 2004 v1. The browser's content area displays the following XML code:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:lom="http://ltsc.ieee.org/xsd/LOM"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
  xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
  xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
  identifier="SCORM2004_31F10Example-1"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
  http://ltsc.ieee.org/xsd/LOM lom.xsd http://www.adlnet.org/xsd/adlcp_v1p3
  adlcp_v1p3.xsd http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd
  http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd
  http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd" version="1.3">
- <metadata>
  <schema>ADL SCORM</schema>
  <schemaversion>CAM 1.3</schemaversion>
  <adlcp:location>xml/caCOURSE_8_metadata.xml</adlcp:location>
</metadata>
- <organizations default="ORG_ID.SCORM2004_31F10Example_1.ORG_1">
- <organization identifier="ORG_ID.SCORM2004_31F10Example_1.ORG_1"
  structure="hierarchical" adlseq:objectivesGlobalToSystem="false">
  <title>31F10 WF#1 Example</title>
  - <item identifier="SCO_2000.SCORM2004_31F10Example_1.INTRO"
    identifierref="R_2000" isVisible="true">
    <title>Introduction</title>
    - <adlnav:presentation>
      - <adlnav:navigationInterface>
        <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
        <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
        </adlnav:navigationInterface>
      </adlnav:presentation>
    - <imsss:sequencing>
      <imsss:controlMode choice="true" choiceExit="true" flow="true"
        forwardOnly="false" useCurrentAttemptObjectiveInfo="true"
        useCurrentAttemptProgressInfo="true" />
      <imsss:rollupRules objectiveMeasureWeight="0" />
    - <imsss:objectives>
      <imsss:primaryObjective />
      - <imsss:objective objectiveID="intro">
        <imsss:mapInfo targetObjectiveID="intro obi" />
      </imsss:objective>
    </imsss:objectives>
  </item>
</organization>
</organizations>
```

Figure 7.2a

7.3 Manifest in Detail

The following describes the manifest in detail with examples of the XML code. Descriptions are in the left column and individual parts of the manifest are contained in the right column.

All identifiers must be unique within the manifest and provided by the author per SCORM® specifications.

<p>This part always stays the same except for the identifier. These attributes declare schemas and namespaces. The physical Schema files are located in the root of the content package.</p>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <manifest version="1.3" identifier="manifest1" xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd http://ltsc.ieee.org/xsd/LOM lom.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3" xmlns:imsss="http://www.imsglobal.org/xsd/imsss" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1" xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3" xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3" xmlns:lom="http://ltsc.ieee.org/xsd/LOM"></pre>
<p>Required metadata designator tags</p>	<pre><metadata> <schema>ADL SCORM</schema> <schemaversion>2004 3rd Edition</schemaversion> </metadata></pre>
<p><organizations> tag describes one or more <organization> structures. The default value identifies which organization to launch for the default.</p>	<pre><organizations default="COURSE_1"></pre>
<p>Title of package goes here</p> <p>This is where the table of contents begins using the <title> tags to create the table of</p>	<pre><organization identifier="COURSE_1" adlseq:objectivesGlobalToSystem="false"> <title>Package Title</title></pre>

<p>contents.</p> <p>PROGRAMMER INFO: SCORM® specifications have the default manifest global objectives mapped to system level. Since the Army only allows mapping to the manifest level, the system level objectives must be set to false. See Army Business Rule S1 for more information on global objectives.</p>	
Dependent SCO	<pre><item identifier="SCO_480" identifierref="R_421"> <title>Introduction to Instruction</title> </item></pre>
Clusters use <item> tag containing other <item> tags. No file resources for clusters	<pre><item identifier="B_154"> <title>Action Statement of the TLO</title></pre>
A SCO. (Uses the <item> tag Notice the identifierref. This <item> references resources (files) that are used in the SCO	<pre><item identifier="SCO_521" identifierref="R_S521"> <title>Introduction to the TLO</title> </item></pre>
Another SCO reference (not expanded below in <resources>)	<pre><item identifier="SCO_520" identifierref="R_520"> <title>ELO Title</title> </item></pre>
Another SCO (not expanded below in <resources>)	<pre><item identifier="SCO_480" identifierref="R_480"> <title>ELO Title</title> </item></pre>
Sequencing code and closing item tag for Cluster	<pre><imsss:sequencing> <imsss:controlMode choice="true"/> </imsss:sequencing> </item></pre>

Content Organization Metadata reference for <organization> (metadata in separate file)	<metadata> <adlcp:location>metadata/ca_course.xml</adlcp:location> </metadata>	
Sequencing code for organization	<imsss:sequencing> <imsss:controlMode choice="true" flow="true"/> </imsss:sequencing>	
Closing tag for Organization	</organization>	
Closing tag for Organizations	</organizations>	
END TABLE OF CONTENTS and BEGINNING OF COURSEWARE PHYSICAL FILE REFERENCES		
Opening tag for Resources	<resources>	
Opening tag for Resource (This is a SCO)	<resource identifier="R_S521" type="webcontent" adlcp:SCORMType="sco" href="unitsafety/index.html">	
Assets for the SCO	File tag for Asset (SCO Launch file must be included in a <file> tag.)	<file href="unitsafety/index.html"/>
	Another file Asset	<file href="unitsafety/us0010.html"/>
	Tag for Asset that this SCO is dependent on	<dependency identifierref="A_137"/>

Closing tag for R_S521 Resource	</resource>
Opening tag for Resource (This is an asset referenced as a dependency.)	<resource identifier="A_137 type="webcontent" adlcp:SCORMType="asset">
File tag for Asset	<file href="images/graphic1.gif"/>
Another Asset	<file href="images/graphic2.gif"/>
Another Asset	<file href="images/movie1.swf"/>>
Closing tag for A_137 Resource	</resource>
Closing tag for Resources	</resources>
Closing tag for Manifest	</manifest>

Figure 7.3a

7.4 Army SCORM® Business Rule S2 Item Analysis



Army Business Rule S2 (SCORM® Item Analysis): Developers of SCO assessments must make use of the SCORM® Interactions Data Model Element to record information about the learner's response for validation purposes. Interactions Data Model Elements Identifier, Type, Correct Responses, Learner Response, and Result are required for each question. The Interaction elements Timed Stamp and Latency are required for timed exams, and posting the Weighting Interaction element is required for an examination that varies the scoring weight for questions. Figure 7.9a presents the Army requirements for each SCORM® Interaction Data Model Element.

7.5 Session Time Example

This value is the amount of time in hours, minutes, and seconds that the learner has spent in the SCO at the time they leave it. That is, this represents the time from beginning of the session to the end of a single use of the SCO.

The JavaScript date object is the vehicle that allows the manipulation of the date object based on a millisecond value, which is then converted back to the form desired. The format is PTnHnMnS where "n" is the numeric value. If a value is zero, then the notation for that unit of time is deleted from the format. For example, if there are no minutes, then the "nM" is deleted from the formatted value. The time format is completely presented in the SCORM® Content Aggregation Model (CAM) specifications document.

Example logic:

```
//format example for sessionTime - "PT23M56S"  
// example shows 23 minutes, 56 seconds  
  
1. Get time the SCO was launched (start time)  
2. Get the time the learner is leaving the SCO (end time)  
3. Subtract the start time from the end time which indicates the  
   session time.  
4. Convert to correct timeinterval data type format  
5. Send session time to the LMS
```

Figure 7.5a

A timer function is needed to track the time the learner spends in the current session of the SCO and send to the LMS upon leaving the SCO. The timer would begin immediately after calling Initialize and stop right before setting the value in the LMS and then Terminate. Because of the stateless nature of HTML, using frames would be one solution to store the time value that the SCO was launched.

Following are JavaScript functions that will start the timer for the beginning value, obtain the ending value and find the difference, and then convert the seconds to hours, minutes, and seconds, and write the formatted value to "cmi.session_time":

```
/* set a global variable on the parent frame */  
var startTime;  
  
/* This function stores the time the SCO was launched in a global  
variable 'startTime'. Call this function after doInitialize */  
function startTimer()  
{  
    startTime = new Date().getTime();  
}  
  
// This function calculates the time learner was in the activity. Call  
this function right before Terminate  
function setSessionTime() {  
  
    var currentTime = new Date();  
    var endTime = currentTime.getTime()  
  
    var calculatedTime = endTime-startTime;  
    var totalHours = Math.floor(calculatedTime/1000/60/60);  
  
    calculatedTime = calculatedTime - totalHours*1000*60*60  
    if ( totalHours < 1000 && totalHours > 99 ) {  
        totalHours = "0"+totalHours.toString();  
    } else if ( totalHours < 100 && totalHours > 9 ) {  
        totalHours = "00"+totalHours.toString();  
    } else if ( totalHours < 10 ) {  
        totalHours = "000"+totalHours.toString();  
    }  
}
```

```

var totalMinutes = Math.floor(calculatedTime/1000/60);
calculatedTime = calculatedTime - totalMinutes*1000*60;
if ( totalMinutes < 10 ) {
    totalMinutes = "0"+totalMinutes.toString();
}

var totalSeconds = Math.floor(calculatedTime/1000);
if ( totalSeconds < 10 ) {
    totalSeconds = "0"+totalSeconds.toString();
}
var sessionTime = "PT";
if (parseInt(totalHours)!=0)
    sessionTime += totalHours + "H";
if (parseInt(totalMinutes) !=0)
    sessionTime += totalMinutes + "M";
if (parseInt(totalSeconds)!=0)
    sessionTime += totalSeconds + "S";

doSetValue("cmi.session_time", sessionTime);
}

```

Figure 7.5b

Another method would be to use "cmi.suspend_data" or "cmi.comments" as a place to store the start time just after Initialize is called. Just before Terminate is called, retrieve this value and calculate the elapsed time.

Code executed right after Initialize:

```

var startTime = new Date().getTime(); // time in seconds
doSetValue( "cmi.suspend_data", startTime );

```

Figure 7.5c

Code executed right before Terminate:

```

var startTime=doGetValue("cmi.suspend_data");

// then use the setSessionTime function shown above

```

Figure 7.5d

7.6 Total Time Example

This is a 'read-only' value. The LMS will initialize total_time to PTOH0M0S the first time the SCO is launched and then use SCO reported values of session_time to keep an accumulated total.

```
variable = doGetValue("cmi.total_time");
```

Figure 7.6a

7.7 Overview

The **Interactions Data Model Element** is provided to allow SCO developers the ability to record the status of a learner's interaction with an individual test item/question. Interactions are learner responses to individual questions in a SCO. The data stored in an interaction element can be used to evaluate the effectiveness of checks on learning or exam questions, either by the SCO or by some courseware data management program provided by the LMS vendor. Each test item/question may have its own Interactions Data Element Model, which is a collection of more than 10 different pieces of information about the learner's response to a test item/question. The SCORM® specification requires LMS vendors to allow for at least 250 of these collections to be stored per SCO. This means that an LMS must allow every SCO in a content package the ability to track learner responses for at least 250 test items/questions. Beyond this minimum, “there is no implied behavior an LMS shall have when interactions are requested to be set, other than storage of the data.”¹

The data within an Interactions Data Model Element is not processed by the LMS in any way. The LMS makes no decisions and takes no actions based on the data stored within an interactions element. In previous versions of the SCORM® specifications, the information written to an interaction could not be read by the SCO, and previously set values could not be overwritten.

For SCORM® 2004, the data in an Interactions Data Model Element is available to the SCO for use. A SCO may read previously set data from an interactions element created by that SCO, update data stored in an interactions element, or write new data to the interactions element. A SCO may NOT read or write to an interactions element created by another SCO. If an LMS provides the capability, an LMS may report on the data stored within the interactions element of the different SCOs in a content package, but this is not required by SCORM®.

¹ 4.2.9 “Interactions” SCORM® 2004 3rd EDITION Sharable Content Object Reference Model Run-Time Environment NOVEMBER 16, 2006 VERSION 1.0

Interaction Element Data Fields	SCORM® or Army?	Program SCORM® Data Element Name
1*) Identifier	SCORM®	cmi.interactions.n.id
2*) Type	SCORM®	cmi.interactions.n.type
3) Objective IDs	Not Required	cmi.interactions.n.objective.n.id
4) Timestamp	Army (for a timed exam)	cmi.interactions.n.timestamp
5*) Correct Responses	Army	cmi.interactions.n.correct_responses.n.pattern
6) Weighting	Army (when questions are weighted differently in an examination SCO)	cmi.interactions.n.weighting
7*) Learner Response	Army	cmi.interactions.n.learner_response
8*) Result	Army	cmi.interactions.n.result
9*) Latency	Army	cmi.interactions.n.latency
10) Description	Not Required	cmi.interactions.n.description

*interaction elements that are mandatory for each student answer. The configuration settings in the Army Multi-log parser are set to validate each of these interaction item is recorded by an examination SCO.

Figure 7.7a

Figure 7.7a shows the Interactions Element Data Model, and 10 different data fields that are stored within each collection for each test item/question. There are more data fields within the Interactions Element Data Model that are not depicted by Figure 7.9a, but those shown are the data fields that the SCO developer may write or update from within the SCO, and are the only ones that are addressed in this section.

7.7.1 SCORM® Required Interaction Data Fields

PROGRAMMER INFO #1: The description field is not required to be stored by the Army but caution must be taken with description to ensure that no more that 250 characters are posted in this data element. The ALMS has been known to throw a database error when more that 250 characters are designated in the description data element. Therefore if an examination is storing the description the code must ensure that no more the 250 characters are being sent to the LMS.

PROGRAMMER INFO #2: The description field **MUST NOT** duplicate any part of the question presented to the learner or link a question to lesson objectives. Under no circumstances should the description be a question stem or distracter, the description must not duplicate text presented to the learner. If used, the description should be a brief description of the question that does not compromise test integrity in any way. The Army preference is that description not be used for performance measuring; credit providing assessments, whether as a pretest or posttest. Of the 10 data fields in these collections, the 'identifier' and 'type' data fields are required by SCORM® to be present whenever the SCO uses the interaction element. The Army interaction date

requirements logically “inherit” the SCORM® requirement and also require identifier and type interaction data fields whenever a student answer is recorded.

Identifier

The 'identifier' must be unique from any other identifier used by other interaction elements in that SCO. That means each test item/question in a SCO must have a unique 'identifier', and each test attempt must be reflected in the “cmi.interactions.[number].id”.

The "cmi.interactions.n.id" is specifically formatted as a long identifier type.

```
doSetValue("cmi.interactions.[number].id", "[long_identifier_type]");
```

Figure 7.7.1a

The Army required long_identifier_type in the interactions id would be specified as:

“test_attempt” + “-“ + long_identifier_type”. This makes a valid Army interaction question id of (assume the entry is for learner test attempt 1, question ‘s3df’):

```
doSetValue("cmi.interactions.0.id", "1-ARMY-GLOBAL-071-33-3401_s3df");
```

Figure 7.7.1b

A second valid Army interaction question id (assume the entry is for learner test attempt 3, question ‘6grx’):

```
doSetValue("cmi.interactions.60.id", "3-ARMY-GLOBAL-071-33-3401_6grx");
```

Figure 7.7.1c

Type

The 'type' data field must be included if there is any use of either the 'correct responses' or 'learner responses' field. This value designates what type of question (multiple choice, true-false, etc.) was delivered to the learner. There is a limited vocabulary for the 'type' field (true_false, multiple_choice, fill_in, long_fill_in, matching, performance, sequencing, numeric, other).

```
doSetValue("cmi.interactions.[number].type", "[vocabulary]");
```

Figure 7.7.1c

The word 'number', designates a counting mechanism that always begins with "0" and then increases by 1.

Example:

```
doSetValue("cmi.interactions.1.type", "multiple_choice");
```

Figure 7.7.1d

PROGRAMMER INFO: The 'number' designated above as "1" can only be used subsequently to using "0".

7.7.2 Army Required SCORM® Interaction Data Fields

Timestamp

The timestamp information field in the interactions element is meant to indicate the point in time at which the test item/question was first given to the learner. This value is set by the SCO. No value will be supplied for the timestamp field by the LMS. It is the SCO's responsibility to set this value. Prior to setting a value for this field, the SCO must first designate the ID field for the interaction element.

This field allows the SCO to indicate the time at which the learner begins working on a test item/question. Once the learner concludes the interaction (answers the item/question), the SCO may then retrieve the timestamp, compare it to the current time, and compute the total amount of time the learner spent working on the question. This field is helpful in determining the latency information field, especially in situations where a learner is allowed to leave the SCO and re-enter while working on the question, but is not the only conceivable way of accomplishing this goal. This information field is also helpful when reviewing test item data for a set of students. Knowing when a learner began a question may be helpful for reviewers examining test item data that relates to time-sensitive material in rapidly changing areas of study. While it is recommended that each question interaction data set record the entry time in the timestamp element, at the time, the Army only requires that timestamp be recorded in timed SCOs.

Correct Responses

This information field defines the correct responses available for a given test item/question. Prior to setting a value for correct responses, the SCO must first define the type information field. The number of correct responses available for an individual question depends on the type of question. A 'true/false' question, for example, will only have one possible correct response, either "true" or "false". While it would be possible to evaluate the correctness of a learner's response to a question without using this field of the Interactions Data Model Element, there would be no way to validate the question from the outside without tediously, manually stepping through the SCO. Therefore, this field is critical for test validation.

The "cmi.interactions.n.correct_responses.n.pattern" value describes the pattern of the correct learner response(s) to the interaction.

Type	Correct Responses Pattern
true_false	true or false
multiple_choice	short_identifier_type[,] short_identifier_type
fill_in	{case_matters="true"} {order_matters="true"} localized_string_type[,] localized_string_type
matching	Pairs of identifiers separated by [,] delimiter. For the pairs, the target and source delimited by [.]. Ex. target[.]source[,] target[.]source
performance	{order_matters}step_name[.] step_answer[,] step_name[.]step_answer
sequencing	short_identifier_type[,] short_identifier_type
likert	short_identifier_type
long-fill-in	{case_matters="true"} localized_string_type
numeric	min[:] max if lower and upper bound. If min and max are the same, then pattern is that number. If no lower bound, [:] max. If no upper bound, min[:]. Min and Max designated as valid real(10,7).

Figure 7.7.2a

Following is a code example for storing the pattern of the correct answer of the test item:

```
doSetValue("cmi.interactions.[number].correct_responses.[sub-  
number].pattern", "[pattern for the correct answer]");
```

Figure 7.7.2b

PROGRAMMER INFO: The [number] must start at zero and increment in steps of one, otherwise the LMS will not properly record and report results.

Following are code examples of different types of questions and setting the pattern of the correct response:

```
// for multiple choice if the correct answer is C
doSetValue("cmi.interactions.0.correct_responses.0.pattern", "c");

// for true_false if the correct answer is true
doSetValue("cmi.interactions.0.correct_responses.0.pattern", "true");

// for fill_in if correct answer is rifle or gun
doSetValue("cmi.interactions.0.correct_responses.0.pattern", "rifle");
doSetValue("cmi.interactions.0.correct_responses.1.pattern", "gun");

// for matching if correct answer is 1c, 2a, 3b, 4d
doSetValue("cmi.interactions.0.correct_responses.0.pattern",
"1[.]c[,]2[.]a[,]3[.]b[,]4[.]d");
```

Figure 7.7.2c

Weighting (required only for weighted questions)

The weighting field is intended to describe the weight given to the test item/question. The specifics of how this field is implemented and used by the SCO are NOT defined by SCORM®. The only restriction on this field is that the data be a real number, limited to no more than seven significant decimal figures. If a SCO developer has assigned different weights to test items/questions, it is necessary to use this field.

```
// set to full 1.0
doSetValue("cmi.interactions.0.weighting", "1.0");
// set to one half .5
doSetValue("cmi.interactions.0.weighting", "0.5");
```

Figure 7.7.2d

Learner Response

The learner response field is intended to represent the answer generated by the user in response to the test item/question. Prior to setting a value for correct responses, the SCO must first define the type. The limits on the value of the learner response depend on the type of question, as defined by the type information field of the Interactions Data Model Element. This field is crucial for test validation.

The "cmi.interactions.n.Learner_response" value is programmed similar to the following:

```
doSetValue("cmi.interactions.[number].learner_response", "[Learner
answer]");
```

Figure 7.7.2e

PROGRAMMER INFO: The [number] must start at zero and increment in steps of one, otherwise the LMS will not properly record and report results.

```
doSetValue("cmi.interactions.10.learner_response", "c");
```

Figure 7.7.2f

Result

The result data field of the interaction element is meant to represent the correctness of the learner response. This field can only be set to one of five values:

Appendix A	"correct"
Appendix B	"incorrect"
Appendix C	"unanticipated"
Appendix D	"neutral"
Appendix E	"[a real number with a precision of seven significant digits]"

Regardless of the value set by the SCO for the result field, the SCO is responsible for passing the correct score for the entire SCO. The LMS will not make any determinations of a learner's success, or score for an assessment SCO, based on the result data field in an interactions element. This field contributes only to test and question validation purposes by persisting learner responses for tracking purposes.

The "cmi.interactions.n.result" value indicates how the system judges the described response. The syntax is shown in the example that follows:

```
doSetValue("cmi.interactions.[number].result", "[vocabulary]");
```

Figure 7.7.2g

The following is a valid working example of an API function call setting the result of the first question, "0" for [number], result to "correct":

```
doSetValue("cmi.interactions.0.result", "correct");
```

Figure 7.7.2h



Best Practice: Results of the test item (correct/incorrect) could be stored in an array using JavaScript and sent to the LMS after scoring the learner performance test using a 'for loop' as shown in the following example:

```
var arResult = new Array();

// storing correct/incorrect in array for each test item
function storeResult(result) {
    var i = arResult.length;
    arResult[i] = result;
}

// send results to LMS after scoring
function sendResults(arResult) {
    for (i=0;i<arResult.length;i++) {
        doSetValue("cmi.interactions." + i + ".result", arResult[i]);
    }
}
```

Figure 7.7.2i

When using Flash to develop a learner performance test, it may be easier to use an array inside of Flash to store the test item results. Sending the Flash array to JavaScript may require converting the array into a string and sending the string to the sendResults function.

Use the split method to store the string back into an array as shown in the following example:

```
/* Question results (correct/incorrect) must be sent to the LMS
   after the learner performance test has been scored */

/* Flash sends a string to a Javascript function with the
   Comma as the string delimiter */

function sendResults(stringList) {

    var resultsArray = stringList.split(",");
    for(i=0;i<resultsArray.length;i++) {
        if (resultsArray[i] ) {
            doSetValue("cmi.interactions." + i + ".result",
resultsArray[i]);
        }
    }
}
```

Figure 7.7.2j

Latency

The latency field is intended to represent the time elapsed between the time the test item/question was given to the learner, and the time the learner answered the question. This field is required by the Army for each question of an Examination. The processing of latency is expected to be a cumulative value when a learner returns to a question. The logic to recognize a latency value exists for a question and therefore the current attempt time will increase an established time is left to individual Army DL developers. The value of this field is limited to the **timeinterval** data type as defined by SCORM®. This value must be set by the SCO.

No value will be supplied for the latency field by the LMS. It is the SCO's responsibility to compute and set this value. This data could be very informative when analyzing test item data. The amount of time learners take to complete an interaction may be an indication of the difficulty of the question.

The latency field of the interaction element has no relation to the "cmi.total_time" data element, or any other element outside the Interactions Data Model Element, in the SCORM® Run-Time Environment. The LMS will make no determination about the timing of a SCO based upon this information field. SCO developers must use other elements within the SCORM® Run-Time Environment to handle 'timed SCOs'.

```
doSetValue("cmi.interactions.[number].latency", "[time]");
```

Figure 7.7.2k

```
doSetValue("cmi.interactions.0.latency", "PT12S");
doSetValue("cmi.interactions.0.latency", "PT5M");
```

Figure 7.7.2f

Two different latency times, the first is 12 seconds and the second is 5 minutes.

7.7.3 Test Item Data Collection Example

Following are code examples of different types of learner scenarios in a learner performance test and the coordinating cmi.interactions data model:

Multiple Choice, Correct Learner Response	doSetValue("cmi.interactions.0.id", "ARMY-071-33-3401-01"); doSetValue("cmi.interactions.0.type", "choice"); doSetValue("cmi.interactions.0.correct_responses.0.pattern", "b"); doSetValue("cmi.interactions.0.learner_response", "b"); doSetValue("cmi.interactions.0.result", "correct"); doSetValue("cmi.interactions.0.latency", "PT45S");
Multiple Choice, Incorrect Learner Response	doSetValue("cmi.interactions.1.id", "ARMY-071-33-3401-02"); doSetValue("cmi.interactions.1.type", "choice"); doSetValue("cmi.interactions.1.correct_responses.0.pattern", "b"); doSetValue("cmi.interactions.1.learner_response", "a"); doSetValue("cmi.interactions.1.result", "incorrect"); doSetValue("cmi.interactions.1.latency", "PT2M45S");
True-False, Correct Learner Response	doSetValue("cmi.interactions.2.id", "ARMY-071-33-3401-03"); doSetValue("cmi.interactions.2.type", "true-false"); doSetValue("cmi.interactions.2.correct_responses.0.pattern", "t"); doSetValue("cmi.interactions.2.learner_response", "t"); doSetValue("cmi.interactions.2.result", "correct"); doSetValue("cmi.interactions.2.latency", "PT15M59S");
True-False, Incorrect Learner Response	doSetValue("cmi.interactions.3.id", "ARMY-071-33-3401-04"); doSetValue("cmi.interactions.3.type", "true-false"); doSetValue("cmi.interactions.3.correct_responses.0.pattern", "t"); doSetValue("cmi.interactions.3.learner_response", "f"); doSetValue("cmi.interactions.3.result", "incorrect"); doSetValue("cmi.interactions.3.latency", "PT1M17S");

Figure 7.7.3a

One type of data that is collected is whether the learner answered the test item correctly or not. This data model is the "cmi.interactions.[item number].result". Concerns have arisen about the possibility of a learner viewing the test item results (correct/incorrect) before completion of the learner performance test. Refer to the previous section Figure 7.7.2g, in the Result example area, for more detail on the result interaction element.

When the learner skips questions and then has to go back to the questions, or whenever a learner returns to question a second time the `cmi.interactions` data should be overwritten using the same "n" number, and the latency field should be incremented by the additional time. However any calculated latency time of less than a second can be ignored except for the initial visit.

7.7.4 Total Time Example

This is a 'read-only' value. The LMS will initialize `total_time` to `PT0H0M0S` the first time the SCO is launched and then use SCO reported values of `session_time` to keep an accumulated total.

```
variable = doGetValue("cmi.total_time");
```

Figure 7.7.4a

7.7.5 Mastery Score Example



Army Business Rule S3 (SCORM® Pass): Graded assessment Sharable Content Objects (SCOs) shall ensure that an appropriate value for the `success_status`, either 'passed' or 'failed' is set in the LMS. Graded assessment SCOs that pass a learner score to an LMS shall declare their mastery score to the LMS using the manifest element `<imsss:minNormalizedMeasure>`.

Furthermore:

- These SCOs shall use the mastery score retrieved from the LMS when reporting the passing grade to the learner.
- These SCOs shall not set the `satisfiedByMeasure` attribute of their corresponding `<imsss:primaryObjective>` element to "false".

SCORM® allows the 'scaled passing score' to be designated outside of the SCO within the sequencing code associated with a SCO on the manifest file. This value can be compared with the 'scaled_passing_score' that the learner achieved on the assessment to determine the learner's success or failure of the assessment. To ensure that a graded assessment SCO stores the appropriate status of a learner's success or failure, the developer should:

- Set the 'scaled_passing_score' using the `<minNormalizedMeasure>` element within the manifest
- Initialize the learner's success status
- Set the learner's score
- Retrieve the learner's success status from the LMS

Setting SCO 'Scaled Passing Score' in the Manifest

To support greater reuse and customization of graded learner assessments, the Army requires that the **'Scaled Passing Score'** be set within the XML manifest file. This allows developers to change the **'Scaled Passing Score'** just by altering one value in the 'imsmanifest.xml' file, rather than reworking a SCO within the original authoring tool to edit the embedded value.

The **'Scaled Passing Score'** is defined in the manifest as the value of the `<imsss:minNormalizedMeasure>` element tag associated with the SCO. This value must be of the XML Data Type `xs:decimal`, and be between -1.0000 and 1.0000, inclusive, with a precision of up to four decimal places.

Following is a code example of a **'Scaled Passing Score'** being set to 80%:

```
<imsss:objectives>
  <imsss:primaryObjective objectiveID="SCO_MS" satisfiedByMeasure = "true">
    <imsss:minNormalizedMeasure>0.80</imsss:minNormalizedMeasure>
  </imsss:primaryObjective>
</imsss:objectives>
```

Figure 7.7.5a

The value set for the `<imsss:minNormalizedMeasure>` within the manifest file will be used by the LMS to initialize the run-time data field, "cmi.scaled_passing_score". This data field may be retrieved by the SCO at any time during a learner's session.

Following is an example of retrieving the scaled passing score from the LMS by the SCO:

```
variable = doGetValue("cmi.scaled_passing_score ");
```

Figure 7.7.5b

7.7.6 Learner Scoring and Success Status

A "mastery score" in SCORM® 2004 is the value that determines the minimum score possible for a learner to successfully pass a graded assessment.

The `<minNormalizedMeasure>` manifest element along with the run-time API data fields "cmi.score.scaled", "cmi.scaled_passing_score", and "cmi.success_status" combine within the LMS to evaluate learner performance.

7.7.7 Initializing Learner's Success Status

The first time, in which a learner enters a graded assessment SCO, the value of the run-time data field, "cmi.success_status", should be "unknown". The SCO developer should test this value to ensure that a correct value is set in case the learner is prematurely disconnected from the SCO. This step is redundant in that the LMS is responsible for setting the "cmi.success_status" to

"unknown" in cases where the "cmi.score.scaled" value is unknown; however, testing for this value will ensure that this SCO is as portable as possible.

```
var = doGetValue("cmi.success_status");
if(var == "passed" || var == "failed")
{
    [Insert business logic here for re-entry of assessment SCO here]
}
else if(var == "unknown")
{
    //Do nothing. Just checking to make sure we had a good value.
}
else
{
    doSetValue("cmi.success_status", "unknown");
}
```

Figure 7.7.7a

Depending on the learning strategy implemented by the courseware, a learner may or may not be allowed to exit and re-enter a graded assessment SCO. If a learner is allowed to re-enter a SCO, and the value for "cmi.success_status" is already set to "passed", then that value shall not be changed. A value of "failed" may or may not be changed, depending on the learning strategy implemented by the courseware.

Setting Learner's Score

The learner's performance is evaluated within the SCO using whatever business logic the developer desires. The "cmi.score.scaled" data model element allows the developer the ability to read or write the raw score ("cmi.score.raw"), the minimum score ("cmi.score.min") or the maximum score ("cmi.score.max") for an assessment SCO. However, the only value that will affect sequencing or the learner's Success Status for a graded assessment SCO is the scaled score ("cmi.score.scaled"). This value must be of the XML Data Type xs:decimal, and be between -1.0000 and 1.0000, inclusive, with a precision of up to four decimal places.

Following is an example of how to set the value for "cmi.score.scaled":

```
// learner's scaled score
var LearnerScore = ".58";

// sending the value to the LMS
doSetValue("cmi.score.scaled", LearnerScore);
```

Figure 7.7.7b

If this value is set, then the Objective Measure Status for the primary objective will be set to "true" and the Objective Normalized Measure for the primary objective will be equal to the set value.

Retrieving Learner's Success Status

The "cmi.success_status" data field is meant to indicate a learner's current status within a graded assessment SCO. Valid values are: "passed", "failed", and "unknown". This field is readable and writable from within a SCO; however, the final value of this data field is ultimately determined by the LMS. Whenever a SCO requests the value of the "cmi.success_status" field, the LMS must determine the value for that field, and then return to the SCO the up-to-date value.

In order to determine the final success status, the LMS will consider the values set for "cmi.scaled_passing_score", "cmi.score.scaled", and "cmi.success_status". If there is a value set for "cmi.scaled_passing_score", then the LMS will determine the value of "cmi.success_status" by comparing the value of "cmi.scaled_passing_score" with "cmi.score.scaled". If the scaled score is greater than or equal to the scaled passing score, the learner's success status computes to "passed", otherwise it computes to "failed".

If "cmi.scaled_passing_score" has been set to a valid value, but there is no valid value for "cmi.score.scaled", then "cmi.success_status" will be "unknown", regardless of whether or not the SCO explicitly sets a value for "cmi.success_status" as described in the [Initializing Learner's Success Status](#) section (7.7.14).

If the learner is allowed to re-enter a test and is not denied re-entry when the learner has a success status of "passed", then the "passed" status must not be changed.

7.7.8 Confusing Success Status with Completion Status

Success status has become confused with the completion status. "Completed" does not mean that the learner passed the test. "Completed" means that the learner has gone through the entire SCO but does not indicate that the learner was successful or unsuccessful. A completion status of "completed" as used in a graded assessment indicates that the learner did not exit out prematurely from the test but saw every test item and every page. The learner can complete the test but still be unsuccessful. A learner can obtain a completion status of "completed" and a success status of "failed".

7.7.9 Completion Status Example

Completion status for all SCOs should be stored in the LMS. Valid values are: "completed", "incomplete", "not_attempted", and "unknown". Following are syntax examples of how 'getting' (reading) and 'setting' (writing) the completion status values are coded:

```
variable = doGetValue("cmi.completion_status");  
doSetValue("cmi.completion_status", "[vocabulary]");
```

Figure 7.7.9a

Following are examples of completion status read from the LMS and written to the LMS:

```
//getting or reading the value from the LMS  
var status = doGetValue("cmi.completion_status");  
  
//setting or writing the value to the LMS  
doSetValue("cmi.completion_status", "incomplete");
```

Figure 7.7.9b

There is confusion regarding the completion status and success status, because some LMSs use completion status as an indicator of success.

7.7.10 GetLastError

'GetLastError' allows the SCO to determine an error condition on the communication to and from the LMS. GetLastError is automatically included in programming code when using an API Wrapper file provided by ADL.

This API function call provides a way of assessing whether or not any given API function call was successful. It returns an error status code resulting from the previous API function call.

```
var retVal = doGetLastError();
```

Figure 7.7.10a

Error codes are listed in the 'Content Aggregation Model (CAM)' specifications document, which is a part of the SCORM® Documentation Suite.

7.7.11 Scaled Passing Score Example

The scaled passing score is indicated in the following example within the <imsss:minNormalizedMeasure> tag as 80%. The value is normalized between -1 and 1, inclusive.

```
<imsss:objectives>  
  <imsss:primaryObjective objectiveID="SCO_MS" satisfiedByMeasure = "true">  
    <imsss:minNormalizedMeasure>0.80</imsss:minNormalizedMeasure>  
  </imsss:primaryObjective>  
</imsss:objectives>
```

Figure 7.7.11a

For additional information, refer to the [Learner Scoring and Success Status](#) section (7.7.7).

7.7.12 Army Validation Business Rules

This section presents the Army Business Rules S4, S5, and S6 – Army mandatory SCORM® Methods and Data Model Elements. Army mandated method calls and data element usage is validated by the Army Log Parser Certification program’s examination of the ADL Certification Test Suite (CTS) log files



Army Business Rule S4 (SCORM® Methods and Data Models Elements Required for Instructional content – Non-scoring SCOs): The Army requires three methods be called by every SCO:

- Initialize()
- Terminate()
- Commit()

Also the Army requires the following SCORM® data model elements be either set for all non-exam SCOs:

- cmi.exit
- cmi.success_status
- cmi.completion_status
- cmi.location
- cmi.session_time

Method calls and data element usage are validated during the Army log parser processing of the Run-Time Environment log file for exam SCOs.



Army Business Rule S5 (SCORM® Methods and Data Models Elements Required for Performance Measure content – Scoring SCOs): The Army requires four methods be called by every scoring SCO:

- Initialize()
- Terminate()
- GetLastError()*
- Commit()

Also, the Army requires the following SCORM® data model elements be set for all scoring SCOs:

- cmi.exit
- cmi.completion_status
- cmi.scaled_score
- cmi.session_time

- cmi.interactions.n.id
- cmi.interactions.n.type
- cmi.interactions.n.correct.response.n.pattern
- cmi.interactions.n.learner_response
- cmi.interactions.n.latency
- cmi.interactions.n.timestamp (for a timed exam)
- cmi.interactions.n.result
- cmi.interactions.n.weighting (when questions are weighted differently in an examination SCO)

Additionally, the SCORM® data model element cmi.success_status and cmi.scaled_passing_score value must be read via the GetValue method for all scoring SCOs. Method calls and data element usage are validated during the Army log parser processing of the Run-time Environment log file for exam SCOs.

* SCORM® GetLastError() method requirements are clarified in Army Business Rule S6



Army Business Rule S6 (SCORM® Error Management): The SCORM® API has three API methods used to handle errors, none affect or alter the error code for the API Instance:

- GetLastError()
- GetErrorString()
- GetDiagnostic()

The Army requires that the GetLastError() method be called from all scoring SCOs and this is validated during the Army Multi-log parser processing of the Run-time Environment log file for a scoring SCO.



Best Practice: Whenever an error is reported, indicated by the APIwrapper function return value “false”; courseware scripting should diagnose the failure and either correct the situation or notify the Learner of a possible malfunction.

The usage of the Error handling is illustrated by the following code snippet from the ADL provided API program doTerminate function and the explanation that follows:

```

/*****
**
** Function doTerminate()
** Inputs: None
** Return: true if successful
** false if failed.
**
** Description:
** Close communication with LMS by calling the Terminate
** function which will be implemented by the LMS

```

```

**
*****/
function doTerminate()
{
if (! initialized) return "true";
var api = getAPIHandle();
if (api == null)
{
message("Unable to locate the LMS's API Implementation.\nTerminate was not
successful.");
return "false";
}
else
{
// call the Terminate function that should be implemented by the API
var result = api.Terminate("");
if (result.toString() != "true")
{
var err = ErrorHandler();
message("Terminate failed with error code: " + err.code);
}
}
initialized = false;
return result.toString();
}

```

Figure 7.7.12a

Upon returning the courseware would read the “false” value returned by the function call and use the `GetLastError()` Method for the specific error value and if necessary use the other two error processing methods, `GetErrorString()` and `GetDiagnostic()`, to correct the error or inform the Learner of the problem. For more information on error codes, messages and corrections, see section 3.1.7, “API Implementation Error Codes”, in the SCORM® 2004 3rd Edition Run-time Environment reference manual, SCORM® RTE.

8. Army Courseware Delivery Business Rules

This section contains normative information in the content of the three Army Business Rules and instructional information that provides general guidance on processes necessary to create SCORM® content packages, descriptions of logical file structures, and illustrative diagrams.

8.1 Packaging and Delivery

SCORM® indicates that the purpose of Content Packaging is to provide a standardized way to exchange digital learning resources between different systems or tools. A Resource Content Aggregation Package, that is, the PIF, contains #1) a manifest file, “imsmanifest.xml”, describing the entire package and #2) all physical files, which provide the support files and the entire learner consumable IMI product.

PROGRAMMER INFO: SCORM® 2004 3rd Edition (or 4th Edition either) does not currently recommend use of (sub)manifests due to confusion on the cases for using (sub)manifests, requirements on syntax, and behavior processing. All previously defined requirements for (sub)manifests have been removed from SCORM® documentation.



Army Business Rule #D1 (Delivery Rules): All SCOs (content and graded assessments) must be contained in a content package. SCORM® Content Packages must contain a manifest (imsmanifest.xml) file and all of the SCORM® and extension schemas in the root of the package.

All physical files required for the courseware must be referenced locally and contained within the content package and disclosed on the imsmanifest.xml file. Metadata is supplied as separate files and these files must be contained within the content package and disclosed on the manifest file.

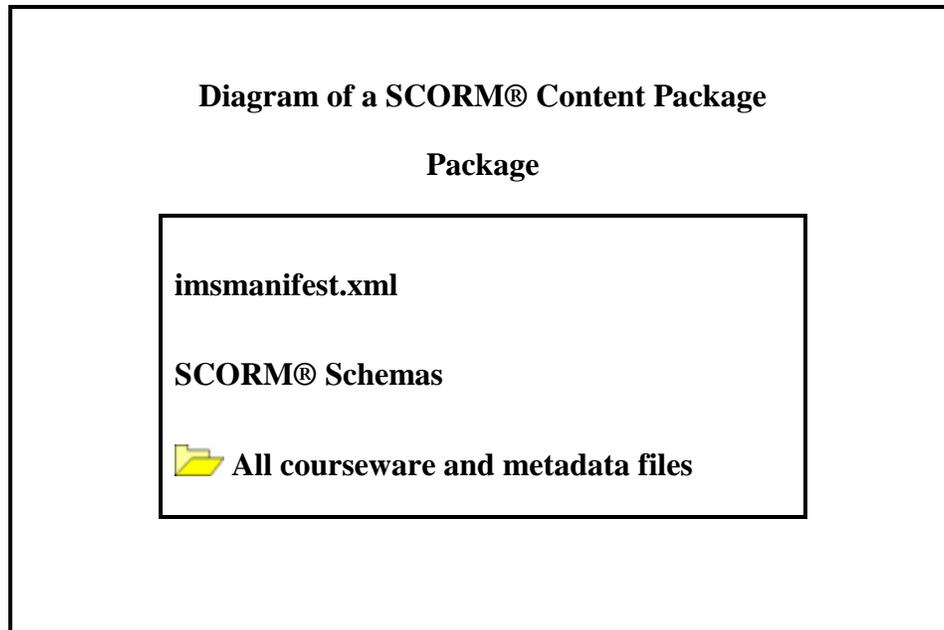


Figure 8.1a

Under SCORM® 2004, the LMS does not require separating out assessments from the content. Instructional strategies are now enforced by the sequencing component of SCORM®.

PROGRAMMER INFO: There are currently no known package size constraints in the Army Learning Management System (ALMS) (Saba 2005). However, anything over 100 MB may have to be loaded by an administrator locally rather than remotely via an Internet connection. This is due to bandwidth issues of loading content remotely, not actual LMS constraints. Content over 100 MB has been successfully loaded. At the time this document was authored, all uploading of content to the production ALMS is done by an administrator of the LMS. Remote Courseware Managers may only upload content to the Common Test Environment (CTE), or development environment.

8.1.1 Creating the Manifest File

The manifest file is the most important file in the package. Specific data on the manifest file is read into the Learning Management System (LMS). The LMS uses this file to determine the table of contents to display and launch file for each SCOs. Content repositories use this file in aggregating and de-aggregating packages. Metadata is determined and accessed by using the information in this file.

The activity tree is a representation of the table of contents, which is XML-tagged within the manifest file, and is created in the LMS when the SCORM® Content Package is launched.

For examples of coding a manifest, refer to the Programming Examples in the Army Best Practice and script examples section titled "[XML Examples on the Manifest](#)".

Two Types of PIFs and two types of Manifest Files:

There are two types of manifest files: Content Aggregation Manifest file and Resource Package Manifest file. A Content Aggregation Manifest contains an <organizations> section, which describes a distinct courseware structure such as a table of contents. A Resource Package does not contain an <organizations> section and, therefore, no table of contents. A Resource Content Package only provides a method of transferring learning resources with supporting files, but does not provide a learning context for the learning resources.

The diagram following shows the differences between the two different types of manifest files:

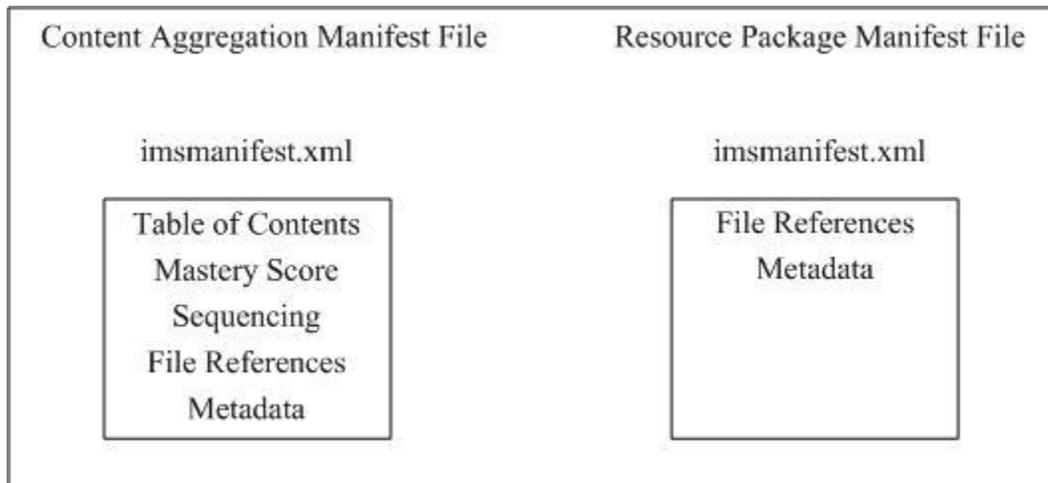


Figure 8.1.1a

8.1.2 How to Create a Content Package per the SCORM® Implementation Guide

Step by step procedure to create a SCORM® Content Package:

1. Determine whether you want to package the files as a PIF. Nearly all credit producing Army DL will be browser-based IMI following SCORM® specifications and therefore delivered as a PIF. All SCORM® content will be packaged and delivered to the government (and distributed to the ALCMC, as one or more PIFs. Deliverables, or more accurately, learner training consumables, not loaded into an LMS; for example, a non-browser based training product such as a modeling and simulation product, an installable file-set is developed for delivery. However, note that the transfer of the installable file-set deliverable to the government is via a Resource Content Package with appropriate metadata files. The non-PIF installable file-set file structure is then distributed by the government.

- PIF – Package Interchange File is a representation of the content package components using the PKZIP Version 2.04g archive format (zip). The PIF provides a concise Web delivery format that can be used to transport content packages between systems (zipped).
 - Non-PIF – a file or file structure that be transported and used on a CD-ROM, DVD, or other file system.
2. Continuing with instructions to create a PIF, place the imsmanifest.xml file at the root level of the package.
 3. Place all schemas referenced by the manifest at the root level of the package. These metadata schemas must also be located in the folder(s) where the metadata resides. In other words, if a folder contains metadata, then the metadata schemas must be copied into this folder.
 4. Place all physical files needed by the packaged courses, lessons, etc., where you refer to them in your manifest.
 5. Compress all files with the imsmanifest.xml file in the root folder and no parent folder using the PKZIP Version 2.04g standard to create one single file. That compressed file is the PIF.
 6. Test the newly created package using the Content Package test of the ADL Conformance Test Suite .
 7. If the content package passes the conformance test, it is ready for delivery to the government a part of a final submission of Army DL for fielding on the ALCMC, or loaded into an LMS for testing, generating learner progress screen captures, or group trials.



Best Practice: It is recommended that the ADL CTS Content Package only option is run as the last step before transferring the PIF to the government to ensure log files unique hash is accurate.

To create a PIF, everything in the large oval shape on the RIGHT side of the following figure would be zipped up using a Microsoft® Windows interface to PKZIP such as WinZip. Notice that the imsmanifest.xml file **must** be in the root of the package.

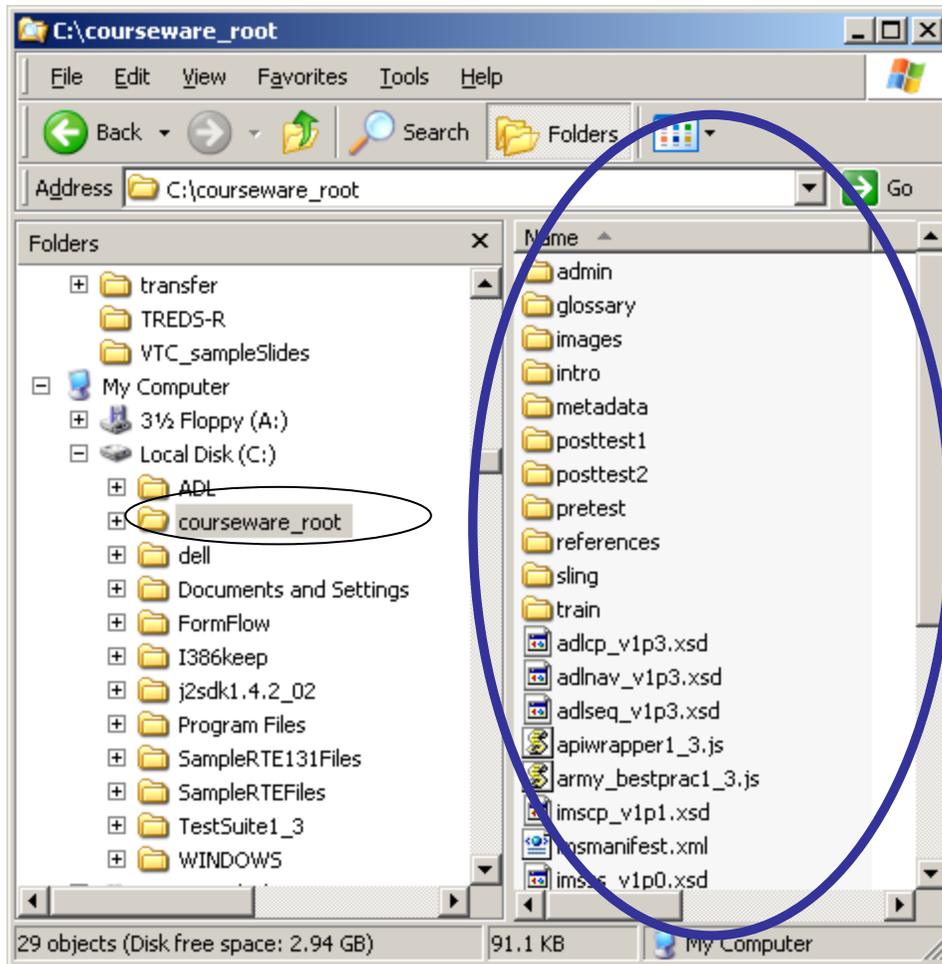


Figure 8.1.2a

WinZip instructions for creating a PIF and other helpful information can be found at <http://www.atsc.army.mil/itsd/imi/document/PIF.doc>.

8.1.3 Final Courseware Packaging and Delivery Requirements

The current guidance on final courseware delivery for Web-based courseware is provided in the USArmy Acceptance Criteria for SCORM® 2004 3rd Edition courseware.

The remaining material in this section is normative information.



Army Business Rule D2 (Delivery Consumables): The Government requires a “Gold Copy” of all eLearning content to be delivered using SCORM® content packaging guidelines. The “Gold Copy” requirement includes all file based content that is consumable by the learner. Furthermore, the files will be packaged into logical bundles and delivered as Package Interchange File (PIF) file(s). Other contract deliverables (for example, original source files, SCORM® testing log files, answer keys, loading instructions, etc.) must be delivered as separate files external to the courseware PIF file(s).

It is important that only the courseware files, metadata files, imsmanifest.xml file, and the schema files (examples shown in Figure 8.1.2a) be included in the PIF. Other files or other contract deliverables should not be included in the PIF..

8.1.4 Army Delivery Documentation Business Rule



Army Business Rule D3 (Delivery Documentation): All SCORM® content submitted to the Army to be hosted on an Army Learning Management System must be accompanied by verification of quality documentation in two areas: 1) Screen captures showing LMS display of Learner progress to completion, normally successful completion and unsuccessful completion for graded examinations, and 2) Each SCORM® Content Package must have SCORM® Certification Test Suite (CTS) log files for the content package conformance test option and each SCO must have the log file from the Run-Time Conformance test option.

The Government expects a logical folder structure and separate DVDs or distinct folders to easily identify contract deliverables. Refer to the most current Task Order for specific packaging contract deliverables.

9. APPENDIX A: Test Item Data (Interactions) Table

Appendix A content is informative.

The following is a suggested test item data table for the developer to fill out for the programmer:

Is this a timed assessment?

<input type="checkbox"/>	Yes
<input type="checkbox"/>	No

Question	Question Type	Correct Responses	Weight (if applicable)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			

Following is a populated example test item data table:

Is this a timed assessment?

	Yes
X	No

Question	Question Type	Correct Response(s)	Weight (if applicable)
1	true/false	true	.25
2	multiple choice	A, C	.50
3	true/false	false	.25
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			

Table A.1

10.APPENDIX B: Metadata Tables

The information in table B.1 is normative information and the information in Table B.2 is normative when duplicating Table B.1, however the sample metadata entries are informative information.

The following is a metadata table that could be used by the training developer to fill out for the programmer:

Metadata Required	Metadata Tag Information for Programmer (from Developer)
Catalog Identifier	ALCMC
Entry Identifier	TBD
Title of Learning Resource	
Language of Learning Resource	
Description of Learning Resource	
Keywords	
Type of Metadata	Choose one: SCO (2) or Collection of SCOs (3) or Content Organization or Package (4)
Version of Learning Resource	
Status of Package Submittal	
Proponent's Role	
Proponent Name School Code Address E-mail	
Date of Submittal	
Metadata Catalog Identifier	TBD
Metadata Entry Identifier	TBD
Metadata Specification Used to Create this Metadata	LOMv1.0 SCORM_CAM_1.3.1 ADLv1.0
Language of Metadata file	
File Format (MIME types) (That is.html, .jpg, .gif, .swf, .avi, .mpeg, etc.)	
Cost of Learning Resource	
Copyright and Other Restrictions	
MOS and Skill Level	

Metadata Required	Metadata Tag Information for Programmer (from Developer)
SQI	
ASI	
Task Numbers and Task Descriptions	
Learning Objectives (Action, Condition, Standard)	
Section 508 Compliant	
Security Level (Foreign disclosure)	

Figure B.1

Following is a populated example of a metadata table:

Metadata Required	Metadata Tag Information for Programmer (from Developer)
Catalog Identifier	ALCMC
Entry Identifier	TBD
Title of Learning Resource	Introduce Basic M60 Machine Gun Training
Language of Learning Resource	English
Description of Learning Resource	Basic instruction on U.S. Army Infantry M60 Machine Gun laying using field expedients, range card preparation, maintenance, function check performance, loading, unloading, malfunction corrections, and target engaging
Keywords	Infantry; M60 laying using field expedients; Notched Stakes
Type of Metadata	2
Version of Learning Resource	1.0 (for final delivery)
Status of Package Submittal	final
Proponent's Role	publisher
Proponent Name	U.S. Army Infantry School
School Code	071
Address	6751 Constitution Loop, Suite 650 Fort Benning, GA 31905-4502
E-mail	soldierinfo-abcxyz@benning.army.mil
Date of Submittal	2004-07-10
Metadata Catalog Identifier	TBD
Metadata Entry Identifier	TBD
Metadata Specification Used to Create this	LOMv1.0

Metadata Required	Metadata Tag Information for Programmer (from Developer)
Metadata	SCORM_CAM_1.3.1 ADLv1.0
Language of Metadata file	English
File Format (MIME types) (That is.html, .jpg, .gif, .swf, .avi, .mpeg, etc.)	Text: .html Image: .gif, .jpeg Applications: Shockwave, Flash
Cost of Learning Resource	no
Copyright and Other Restrictions	no
MOS and Skill Level	11C2 Indirect Fire Infantryman
SQI	E Mountaineer
ASI	Q6 Long Range Surveillance Leader
Task Numbers and Task Descriptions	071-312-3003 Lay An M60 Machine Gun Using Field Expedients 071-312-3007 Prepare A Range Card For An M60 Machine Gun 071-312-3025 Man An M60 Machine Gun
Learning Objectives (Action, Condition, Standard)	Action: Lay An M60 Machine Gun Using Field Expedients Condition: Given Interactive Multimedia Instruction Standard: The Standards are met when the learner has completed the IMI lesson and achieved a passing score on a separately administered test.
508 Compliant	Not 508 Compliant
Security Level (Foreign disclosure)	FD1

Figure B.2

11.APPENDIX C: SCORM® Data Types

The following table should be considered informative information, however it is intended to support the normative source of this data. Normative information for SCORM® Data Types is found in “SCORM 2004 3rd Edition Run-Time Environment (RTE) Version 1.0”, Section 4.1.1.7 Data Types.

SCORM® Data Type	Definition of Data Type
Characterstring	A string of characters that are defined in ISO 10646, which is equivalent to the Unicode Standard.
Language Type	<p>A data type used to represent a language. The format of this data type is a characterstring consisting of a required language code (langcode) followed by multiple, optional, hyphen-prefixed subcodes (subcode):</p> <pre>language_type ::= langcode ["-" subcode]*</pre>
Localized string type	<p>A localized characterstring is a characterstring that has an indicator of the language of the characterstring. There are certain data model elements where the language information is important. SCORM® applies a reserved delimiter for representing the language of the characterstring: {lang=<language_type>}. The format of the characterstring is required to have the following syntax:</p> <pre>"{lang=<language_type><actual characterstring>"</pre> <p>Example: “{lang=en} The content presented an excellent point dealing with the topic.”</p>
Long Identifier Type	A unique characterstring that conforms to the syntax defined for Universal Resource Identifiers (URI), refer to Request for Comments (RFC) 2396.

SCORM® Data Type	Definition of Data Type
Short Identifier Type	A unique characterstring that conforms to the syntax defined for Universal Resource Identifiers (URI), refer to RFC 2396.
Integer	A member of the set of positive whole numbers, negative whole numbers and zero.
State	Some data model elements have a defined set of states. Each state is found to a reserved token. These are further defined in the Data Model Element Implementation Requirements section in each data model Run-Time Environment explanation.
Real	The real(10,7) data type denotes a real number and uses a paired set for system and significant digits. For example real(10,7) is a decimal number with seven significant digits.
time (second, 10, 0)	<p>The format of the characterstring shall be as follows:</p> <p>YYYY[-MM[-DD[Thh[:mm[:ss[.s[TZD]]]]]]]</p> <p>where</p> <ul style="list-style-type: none"> • YYYY: A four-digit year • MM: A two-digit month • DD: A two-digit day of month • hh: Two-digits of hour • mm: Two-digits of minute • ss: Two-digits of second • s: One 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. • TZD: Time zone designator (Z for UTC or +hh:mm or -hh:mm). • At least the four-digit year must be present. If additional parts of the time are included, the character literals -, T, : and . are part of the character lexical representation.

SCORM® Data Type	Definition of Data Type
	<p>Examples:</p> <ul style="list-style-type: none"> • 2005 • 2005-07-25T03:00:00 • 2005-07-25T03:00:00-03:10 • 2005-07-25T03:30.35+05 • 2005-07-25T03:00:00Z
timeinterval	<p>The timeinterval (second, 10, 2) denotes that the value for the data model element timeinterval represents elapsed time with a precision of 0.01 seconds</p> <p>The format of the characterstring shall be as follows:</p> <p>P[yY][mM][dD][T[hH][nM][s[.s]S]] where:</p> <ul style="list-style-type: none"> • y: The number of years • m: The number of months • d: The number of days • h: The number of hours • n: The number of minutes • s: The number of seconds or fraction of seconds • The character literals designators P, Y, M, D, T, H, M and S shall appear if the corresponding non-zero value is present. <p>Example:</p> <ul style="list-style-type: none"> • P1Y3M2DT3H indicates a period of time of 1 year, 3 months, 2 days and 3 hours • PT3H5M indicates a period of time of 3 hours and 5 minutes

Figure C.1

12.APPENDIX D: Glossary and Acronyms

The information in Appendix D is informative.

Acronyms/Terms	Definitions
AAR	After Action Review
ADL or The ADL Initiative	Advanced Distributed Learning. The ADL Initiative, sponsored by the Office of the Secretary of Defense (OSD), is a collaborative effort between government, industry and academia to establish a new DL environment that permits the interoperability of learning tools and course content on a global scale. ADL has developed the SCORM®.
AICC DL	Aviation Industry Computer-Based Training Committee Distributed Learning
ALCMC	Army Learning Content Management Capabilities
ALMS	The Army's Learning Management System provides a suite of capabilities that integrate training and education support capabilities and enhance training and education. It is a Web-based information system that delivers training to Soldiers, manages training information, and provides training collaboration, scheduling, and career planning capabilities in both resident and non-resident training environments. It consists of Saba® Enterprise 5.3 Learning Suite, Blackboard® Academic Suite, CENTRA, and Adobe® Acrobat Connect.
API	Application Programming Interface. The API enables the communication of data between content and the RTE provided by an LMS. The use of a common API fulfills many of SCORM®'s high-level requirements for interoperability and reuse.
AR(s)	Army Regulation(s)
ASI	Additional Skill Identifier

Acronyms/Terms	Definitions
Asset	An asset is the smallest, atomic, meaningful learning content object used to develop Web-based IMI training to include the raw media (separate and distinct instructional text, audio, video, graphic, animation, etc.) used to create a Sharable Content Object (SCO). Also referred to as a 'resource'.
ATRRS	Army Training Requirements and Resources System. Legacy system that tracks quota and non-quota managed training
ATSC	Army Training Support Center
Block	A term used in a previous SCORM® standard (SCORM® v1.1), which maps to a Content Aggregation in SCORM® v1.2 and a cluster in SCORM® 2004.
CAM	Content Aggregation Model. One of the SCORM® 2004 Reference manuals books provided on ADL's Web site.
CBT	Computer-based Training
Cluster	 ADL reference: An activity and its children, but not the descendants of its children; activity that contains other activities.
CMI	Classified Military Information
Content Aggregation	A collection of SCOs.
Content SCO	A SCO that does not contain a graded assessment.
Context-neutral Content	In SCORM®, content which can be separated from its SCORM® package and still be considered complete. You can maintain context-neutrality by not referring to other SCOs, avoiding direct links to other SCOs or portions of the content, etc.
Context-specific	Specific to the context of a particular structure or situation.
CUI	Controlled Unclassified Information
Data Model	A set of data elements that ensures a defined set of information about SCOs can be tracked by different LMSs.

Acronyms/Terms	Definitions
Dependent	Relying on another for support, aid, etc; Determined or conditioned by another.
Dependent SCO	A SCO that is 'not independent' and is context-specific and does not have a high degree of reusability (for example, an Introduction SCO for a module or group of lessons). A SCO with context which relies on external content or a particular courseware structure or specific courseware sequence.
DL	Distributed Learning
DOM	Document Object Model
ELLC	Enterprise Lifelong Learning Center
ELO	Enabling Learning Objective
Embedded Learner Performance Test	A graded assessment contained in the same SCO as learning content.
External References	Content that the learner is not tested. Documents that would invite further study, research, or information on the subject matter.
FD	Foreign Disclosure
FM(s)	Field Manual(s)
GFI	Government Furnished Information
Graded Assessment	The act of evaluation resulting in a score or grade.
Granularity	'Becoming granular' or 'showing a granulated structure'. SCORM® recommends small particles by reducing the size of learning objects for reusability and enhancing the learning experience.
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICW	Interactive Courseware

Acronyms/Terms	Definitions
IEEE	Institute of Electrical and Electronics Engineers (pronounced eye-triple-e) is responsible for the standardized advancement of technology related to electricity.
IMDP	Instructional Media Design Package
IMI	Interactive Multimedia Instruction
imsmanifest.xml	An xml file that is stored in the root of a content package that describes the entire package.
Independent SCO	A SCO that teaches a complete learning thought. An independent SCO must not be dependent on any other learning content, must be 'context independent' and must stand alone.
ISO	International Organization for Standardization is an international standard-setting body composed of representatives from various national standards organizations. Link: http://en.wikipedia.org/wiki/Standards_organizations
Launchable Asset	An activity can be developed as a launchable asset when learner tracking is unnecessary.
Leaf Activity	In a SCORM® activity tree, activity that does not consist of another activity. Leaf activity contains content and no children.
Learner Performance Test	Any graded assessment of learner performance such as a pretest or posttest.
Learning Content	Content that is germane to the instruction and on which the learner will be tested.
LCMS	Learning Content Management System
LCO	Learning Content Object
Learning Steps	A learner activity that leads toward achievement of a learning objective. Learning steps are determined when the objective is broken down into its component parts. Often an explicit hierarchical relationship consisting of a terminal learning objective, an enabling learning objective, and learning step.

Acronyms/Terms	Definitions
LOM	Learning Object Metadata. The root element or the beginning of the SCORM® metadata XML record, for example <lom>.
LMS	Learning Management System
Manifest	A common name that refers to the imsmanifest.xml file, which is an essential part of all SCORM® Content Packages. This xml file is stored in the root of a SCORM® Content Package and describes the contents of the entire package.
Menu	A list of available topics or available branching within a SCO.
Metadata	In SCORM®, data about the resources contained in Web-based courseware; used for discoverability. Metadata is often referred to as “data about data”.
MIME	Multipurpose Internet Mail Extensions
MOS	Military Occupational Specialty
Non-learning content	Content that does not teach, but may inform, direct, explain, or provide understanding.
Objective Definitions Page	A page within the SCO that displays the Learning Objectives Actions, Conditions, and Standards.
Offline Player	An application that plays SCORM® conformant courseware outside of a SCORM® conformant LMS.
Pam(s)	Pamphlet(s)
PIF	Package Interchange File. This file is a representation of the content package components using the PKZIP Version 2.04g archive format (zip).
POI	Program of Instruction
Reg(s)	Regulation(s)
Repository	A digital data storage warehouse where learning objects may be accumulated and cataloged for broad distribution and use.
Resource(s)	Files required for the courseware to play. These files can consist of HTML, Flash, graphics, audio, or video files.

Acronyms/Terms	Definitions
RFC(s)	The Request(s) for Comment(s) form an official series of notes, started in 1969, about the Internet. The notes discuss many aspects of computer communication, focusing on networking protocols, procedures, programs, and concepts but also including meeting notes, opinions, and sometimes humor.
Rollup	A single result from many.. A rollup of completion status is a determination of all sub-activities of a cluster. SCORM® implements rollup control in a content package using manifest tags such as rolluprules and conditions.
RTE	Run-Time Environment. One of the SCORM® 2004 Reference manuals books provided on ADL's Web site.
Schema	An XML file that contains programming rules and allowable XML tags that define the structure and content of other XML files.
SCO	Sharable Content Object. A SCO must reference a specific launch URL that utilizes the SCORM® Run-Time Environment (RTE) to communicate with LMSs. A SCO is independent of context, or context-neutral content. A SCO is able to be reused; moved from one content package to another content package without loss of fidelity or instructional value.
SCORM®	Sharable Content Object Reference Model
SME	Subject Matter Expert
SOW	Statement of Work
SQI	Skill Qualification Identifier
SS&N	Simple Sequencing and Navigation. One of the SCORM® 2004 Reference manuals provided on ADL's Web site.
Sustainment	Sustainment is a term used to indicate that the learner re-accesses the learning content after the learner has completed a graded assessment.
TADLP	The Army Distributed Learning Program
TADLP C&I	The Army Distributed Learning Program Capabilities and Implementation
TBD	To Be Developed

Acronyms/Terms	Definitions
TLO	Terminal Learning Objective
TOC	Table of Contents. The main list of available learning content for a course, phase, module, or lesson that describes the structure and behavior of the content. In SCORM®, the TOC is defined in the manifest file and can only show the structure of a package.
TR(s)	TRADOC Regulation(s)
TRADOC	Training and Doctrine Command
TSP	Training Support Package
TWG	Technical Working Group
UI	User Interface
URI	Uniform Resource Identifier. A URI is a compact sequence of characters that provides a simple and extensible means for identifying a resource.
URL	Uniform Resource Locator
URN	Uniform Resource Name
vCard	The generic term for an electronic, virtual information card that can be transferred between computers, PDAs, or other electronic devices through telephone lines, or e-mail networks, or infrared links. vCard is a metadata tag required by the Army.
WRT	with respect to
www	World Wide Web
XML	eXtensible Markup Language

Table D.1