
The Army University

The Army Distributed Learning Program Directorate (DDL TADLP)

Version 1.0

1 May 2019
Purpose
The purpose of the Army Acceptance Criteria for Shareable Content Object Reference Model (SCORM) 2004 3rd Edition Computer-Managed Instruction (CMI) is to specify quality control (QC) and quality assurance (QA) procedures that Contractors and/or Army Activity developers (hereafter referred to as the developer) shall use to verify that CMI, referred to as courseware for short, is compliant with the SCORM 2004 3rd Edition specification. Compliance includes meeting playability requirements and properly executing the Army activity’s instructional strategy prior to submission to the Government for technical acceptance testing. These criteria apply to courseware delivered to the Government in support of The Army Distributed Learning Program (TADLP), as well as “in-house” TADLP courseware developed by the Government.

Organization
The information in this document is organized into four sections with additional appendices as follows:

- **Section 1**: Executive summary and overview of Government testing and review process for SCORM 2004 3rd Edition courseware.
- **Section 2**: Compliance testing procedures for SCORM 2004 3rd Edition courseware that developers shall follow prior to delivery to the Government, as specified in the Army business rules and best practices for SCORM 2004 computer-managed instruction (CMI).
- **Section 3**: Playability testing procedures for SCORM 2004 3rd Edition courseware that developers shall conduct prior to delivery to the Government.
- **Section 4**: Description of the three phases of the Government review process for SCORM 2004 3rd Edition courseware.
- **Appendices**: Detailed information on compliance testing software, testing procedures, and example courseware compliance documentation required for submission to the Government.

Specifications
This document refers to the following specifications for TADLP courseware:

- **Army Business Rules and Best Practices (BRBP) for SCORM 2004 CMI**: All references to the Army BRBP for SCORM 2004 CMI refer to the document located at: [https://tadlp.github.io/brbp/](https://tadlp.github.io/brbp/).


# Table of Contents

1  Executive Summary......................................................................................................................... 1

1.1  Courseware Developer QA/QC Testing.................................................................................................................. 1

1.2  Government Acceptance Testing and Verification ................................................................................................. 1
   1.2.1  IMDP Wireframe Verification........................................................................................................................ 1
   1.2.2  Course Module Compliance Testing ........................................................................................................ 1
   1.2.3  Final Acceptance Review.................................................................................................................................... 1

2  SCORM Compliance Testing Process............................................................................................................. 2

2.1  Criteria for SCORM Compliance Testing ........................................................................................................... 2
   2.1.1  Compliance Testing Software .......................................................................................................................... 2
   2.1.2  Log Files............................................................................................................................................................ 3

2.2  Developer Preparation for Testing.................................................................................................................... 5

2.3  SCORM Compliance Testing Process.............................................................................................................. 5

3  Courseware Playability Testing....................................................................................................................... 7

3.1  General Requirements ........................................................................................................................................ 7

3.2  IMDP Wireframe Test Requirements .................................................................................................................. 7
   3.2.1  Wireframe Testing Procedure......................................................................................................................... 7

3.3  Developer Preparation for Testing.................................................................................................................... 9

3.4  Module Incremental and Final Courseware Playability Testing ........................................................................ 9

3.5  Delivery Platform Courseware Configuration.................................................................................................. 11

3.6  BRBP Checklist for SCORM 2004 CMI.............................................................................................................. 11

3.7  Critical Errors..................................................................................................................................................... 11

3.8  Section 508 Compliance Verification ................................................................................................................ 12

4  Government Review Process .......................................................................................................................... 12

4.1  Instructional Media Design Package Review ..................................................................................................... 12

4.2  Incremental Module Testing ............................................................................................................................... 12

4.3  Final Courseware Approval .............................................................................................................................. 13

Appendix A – Workstation Configuration........................................................................................................... 1

Appendix B – Understanding the ADL Test Suite ................................................................................................. 1

Appendix C – The SCORM Resource Validator .................................................................................................. 1
1 Executive Summary
This section provides a summary of the process that the developer and the Government use for testing and verifying courseware found in this document.

1.1 Courseware Developer QA/QC Testing
Prior to submitting courseware deliverables to the Government for acceptance testing and compliance verification, the developer shall perform SCORM compliance testing and playability testing following the procedures in this document. Courseware developers shall employ QA measures throughout the entire design and development process and conduct QC testing to ensure that courseware meets its intended purpose(s) and is free of errors prior to submitting it to the Government. After successfully completing SCORM and playability testing, the developer shall provide deliverables to the Government in accordance with Attachment 3 of the BLDO Template (02 Mar 2009).

1.2 Government Acceptance Testing and Verification
The Army employs an incremental testing and verification process whereby courseware is reviewed for SCORM compliance and playability at multiple points throughout the entire development phase. The testing and verification process has three main components:

- Instructional Media Design Package (IMDP) Wireframe Verification
- Course Module Compliance Testing
- Final Acceptance Review

1.2.1 IMDP Wireframe Verification
The testing and verification process begins with the developer submitting an IMDP wireframe that acts as a technical proof-of-concept demonstrating the sequencing and navigation required by the instructional design documented in the IMDP. The wireframe does not contain instructional content. The Government will review the wireframe for functionality to verify that it executes the Army activity’s instructional strategy as outlined in the course map in the IDMP.

1.2.2 Course Module Compliance Testing
After the Government verifies that IMDP wireframes meet requirements, the developer begins assembling and packaging courseware modules. A module is any lesson, assessment, or other content that will be presented to the learner as a discrete item – for SCORM courseware, a module is SCORM content package. The Government will conduct incremental compliance testing on modules as they are produced by the developer. The developer shall correct critical issues identified during testing, as well as other deficiencies at the discretion of the Army activity, using an iterative development process.

1.2.3 Final Acceptance Review
The Government will conduct an acceptance review of the final courseware once all modules have undergone compliance testing and been determined by the Government to meet requirements for delivery. The Government will verify that courseware is SCORM-compliant and implements the Army activity’s instructional strategy on the target delivery platform, typically a Learning Management System (LMS) or Learning Content Management System (LCMS).
2 SCORM Compliance Testing Process

This section outlines the criteria that developers shall use to conduct SCORM compliance testing for all courseware deliverables.

2.1 Criteria for SCORM Compliance Testing

The developer shall conduct compliance testing to verify that courseware meets the following requirements:

a. All SCORM content package manifest and metadata XML files are compliant to the SCORM 2004 3rd Edition specification.

b. All SCORM content package metadata files that contain Army metadata tags are compliant with Army requirements.

c. All files in a SCORM content package are listed as resources files package’s manifest in compliance with Army requirements.

d. All Shareable Content Objects (SCOs) utilize the SCORM 2004 3rd Edition Application Programming Interface (API) for communication of learner tracking data to the delivery platform run-time environment in compliance with the SCORM 2004 3rd Edition specification.

e. All SCOs meet Army requirements for SCORM run-time environment behavior utilizing the SCORM 2004 3rd Edition API.

2.1.1 Compliance Testing Software

The Government provides two pieces of software that developers shall use to conduct SCORM compliance tests on all SCORM 2004 3rd Edition courseware:

- **ADL SCORM 2004 3rd Edition Conformance Test Suite**: ADL provides a conformance test suite application for testing SCORM 2004 3rd Edition content against the specification. The test suite allows courseware developers to conduct automated tests on SCORM 2004 3rd Edition content to produce reports of conformance. The latest version of the test suite is 1.1.2 and it requires the Microsoft Internet Explorer 11 web browser and the latest 32-bit Java Run-time Environment (JRE) installed to run.

- **SCORM Resource Validator**: The Army provides the SCORM Resource Validator to replace older applications designed to test SCORM 2004 3rd Edition courseware for Army requirements. The SCORM Resource Validator allows courseware developers to conduct automated tests on SCORM 2004 3rd edition content to produce reports of compliance with Army requirements. The latest version of the SCORM Resource Validator is 2.2 and it requires the Microsoft Windows 7 or 10 operating system to run.
All SCORM 2004 3rd Edition courseware deliverables must be validated by the developer using compliance testing software, and all reports generated produced must be delivered to the Government (see Figure 1).

<table>
<thead>
<tr>
<th>Compliance Testing</th>
<th>During Development</th>
<th>Final Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADL SCORM 2004 3rd Edition Conformance Test Suite</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Content Package Conformance Test</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- Manifest validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Metadata validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Run-time validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SCORM Resource Validator</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Army metadata validation</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- Army SCORM run-time validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Army SCORM content resource validation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1: SCORM 2004 3rd Edition Testing Requirements Matrix**

2.1.2 Log Files
The ADL SCORM 2004 3rd Edition Conformance Test Suite and the SCORM Resource Validator report the results of their automated tests in log files that the developer shall deliver to the Government.

2.1.2.1 ADL SCORM 2004 3rd Edition Conformance Test Suite Logs
The developer shall run the Content Package Conformance Test in the ADL test suite on each SCORM content package produced. The test suite takes a SCORM 2004 3rd Edition content package as an input and validates the content package manifest, any metadata files, and the conformance of Shareable Content Object (SCO) run-time behavior to the SCORM 2004 3rd Edition specification.

The Content Package Conformance Test will produce the following log files:

- **Content Package Summary Log**: An XML log file that contains a summary of conformance results from individual tests (content package, metadata and run-time environment).
- **Content Package Detailed Log**: An XML log file that contains detailed results of conformance tests performed on the content package itself based on the package’s construction.
- **Metadata Detailed Log**: An XML log file for each metadata file detected in the package manifest that contains detailed results of conformance tests performed on the metadata file.
- **SCO Detailed Log**: An XML log for each SCO defined in the content package manifest that contains detailed results of conformance tests performed on the run-time behavior and communication through the SCORM API.

The developer shall deliver all log files produced by the ADL test suite to the Government. See Figure 2 for examples of ADL test suite log file names.
### ADL SCORM 2004 3rd Edition Conformance Test Suite Logs

<table>
<thead>
<tr>
<th>Log Name</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Package Summary Log</td>
<td>2018-05-09_09.56.36584-PACKAGE_SummaryLog.xml</td>
</tr>
<tr>
<td>Content Package Detailed Log</td>
<td>2018-05-09_09.56.36541-PACKAGE_DetailedLog.xml</td>
</tr>
<tr>
<td>Metadata Detailed Log</td>
<td>2018-05-09_09.56.42778-MD1_DetailedLog.xml</td>
</tr>
<tr>
<td>SCO Detailed Log</td>
<td>2018-05-09_09.57.35946-SCO_DetailedLog.xml</td>
</tr>
</tbody>
</table>

![Figure 2: ADL SCORM 2004 3rd Edition Conformance Test Suite Log Files](image)

See Appendix B for further details such as log file examples, and an overview of ADL test suite workflow.

#### 2.1.2.2 SCORM Resource Validator Logs

The developer shall run the SCORM Resource Validator on each SCORM content package and the log files produced for that package from the ADL test suite. The Resource Validator takes a SCORM 2004 3rd Edition content package and set of ADL test suite logs as input and validates the content and log conformance to Army requirements.

The SCORM Resource Validator will produce the following log files:

- **Army Run-Time Validation Log**: An HTML file that contains the results of tests performed by parsing the content of ADL test suite log files for Army requirements.

- **Army Metadata Validation Log**: A DOC file that contains the results of conformance tests run on metadata files detected in the package manifest.

- **Content Package Resource Validation Logs**: A set of four HTML files that contain the results of conformance tests for content package resources:
  
  - **Manifest Resource Log**: A list of files referenced in the content package manifest as resources.
  
  - **Content Package Resource Log**: A list of files detected within the content package itself.
  
  - **Manifest-to-Content-Package Resource Log**: A list of files found in the content package manifest but not found in the content package itself.
  
  - **Content-Package-to-Manifest-Resource Log**: A list of files found in the content package itself but not referenced in the manifest file.

The developer shall deliver all log files produced by the SCORM Resource Validator to the Government. See Figure 3 for SCORM Resource Validator log file names.

### SCORM Resource Validator Logs

<table>
<thead>
<tr>
<th>Log Name</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Army Run-Time Validation Log</td>
<td>ParsedOutput.html</td>
</tr>
<tr>
<td>Army Metadata Validation Log</td>
<td>ValidateMD.doc</td>
</tr>
<tr>
<td>Manifest Resource Log</td>
<td>manifest_files_found.html</td>
</tr>
<tr>
<td>Content Package Resource Log</td>
<td>packaged_files_found.html</td>
</tr>
<tr>
<td>Manifest-to-Content-Package Resource Log</td>
<td>manifest_files_missing.html</td>
</tr>
<tr>
<td>Content-Package-to-Manifest-Resource Log</td>
<td>packaged_files_missing.html</td>
</tr>
</tbody>
</table>

![Figure 3: SCORM Resource Validator Log Files](image)
See Appendix C for further details such as log file examples, and an overview of SCORM Resource Validator workflow.

### 2.2 Developer Preparation for Testing
Prior to conducting conformance testing on SCORM 2004 3rd Edition courseware, the developer shall ensure that the following requirements are met:

- The developer has a clear determination and understanding of the Army requirements for SCORM 2004 3rd Edition courseware according to the Delivery Order.
- The developer has a computer workstation configured for testing with Army SCORM compliance testing software in accordance with Appendix A.
- The developer has verified that the courseware to be tested is the most current version of the courseware that will be sent to the Government as a deliverable.

### 2.3 SCORM Compliance Testing Process
The developer shall conduct compliance testing on SCORM 2004 3rd Edition courseware using the procedures shown in this section. Only after courseware successfully passes all of the conformance tests performed by compliance testing software shall the developer submit log files and courseware content packages to the Government for review. The developer shall perform the following steps for each SCORM 2004 3rd Edition content package:

2. If all ADL test suite tests show the package is conformant, proceed to step 4.
3. Correct deficiencies identified by ADL test suite tests and repeat step 1.
4. Run the SCORM Resource Validator on the content package and previously generated conformant ADL test suite logs.
5. If all SCORM Resource Validator tests show the package and logs are conformant, proceed to step 7.
7. Package all conformant log files from ADL test suite and SCORM Resource Validator for delivery to the Government.

Note that any modification made to content package files at any stage means that the content package has to be tested for conformance using the ADL test suite to ensure that modifications did not introduce additional errors.

Figure 4 on the next page depicts the SCORM compliance testing process that the developer shall follow for each content package.
Figure 4: SCORM Compliance Testing Process
3 Courseware Playability Testing
This section outlines the criteria that developers shall use to conduct target delivery platform playability testing for all courseware deliverables.

3.1 General Requirements
The developer shall conduct playability testing on the target delivery platform to ensure that courseware meets the following requirements:

a. All SCORM content packages execute the Army activity’s instructional strategy as demonstrated by validated wireframe versions of content.
b. All SCORM content packages pass incremental Army playability tests prior to delivery for Government review.

3.2 IMDP Wireframe Test Requirements
The developer shall test wireframe versions SCORM content on the target delivery platform to demonstrate the capability to execute the Army activity’s instructional strategy as documented in the Instructional Media Design Package (IMDP). Executing the Army activity’s instructional strategy entails content successfully:

- Facilitating sequencing and navigation between content on the target delivery platform.
- Following course module flow as outlined in the course map.
- Granting the learner credit for successful completion for each modules, and the course as a whole, on the target delivery platform.

3.2.1 Wireframe Testing Procedure
To verify the functionality of wireframes, the developer must:

- Load and configure SCORM content packages on the target delivery platform testing server in accordance with content configuration guidance for the delivery platform.
- Create course module flows using SCORM content on the target delivery platform testing server that follow all instructional paths laid out in the course map.

Wireframe functionality requirements are satisfied when all learner progress paths are demonstrated via multiple enrollments and each progress path is verified as correct based on learner completion data recorded on the delivery platform. The developer shall test every significant path through the courseware in order to validate its playability against the instructional strategy.

Figure 5 on the next page depicts a sample course module flow diagram. Actual course module flow diagrams shall include the actual titles of SCOs contained in the courseware wireframe.
Figure 5: Generic sample course module flow diagram
3.3 Developer Preparation for Testing
Prior to conducting playability testing of SCORM 2004 3rd Edition courseware, the developer shall ensure that the following requirements are met:

- The developer has verified that the courseware to be tested is the most current version of the courseware that will be sent to the Government as a deliverable.
- The developer has created and provided to the Government a course map and course module flow diagram as part of the IMDP for use in testing wireframe and developed versions of courseware.
- The developer has created and provided to the Government clear answer keys that include at least questions and correct answers for review of graded assessments, checks-on-learning, simulations, and practical exercises.

3.4 Module Incremental and Final Courseware Playability Testing
The developer shall conduct playability reviews of all incremental and final SCORM courseware deliverables to verify that the courseware functions as intended. Playability reviews include, but are not limited to, verification of:

- Correct functionality for all navigation and instructional features (e.g. navigation buttons, video, audio, animation, etc.)
- Accurate scoring and interaction data reporting via SCORM for all learner performance measurement instruments and checks (e.g. checks-on-learning, practical exercises, simulations, and exams).
- Compliance with Section 508 requirements in accordance with the Delivery Order
- Accurate reporting of completion and success statuses to the delivery platform via SCORM.

The developer must ensure that courseware grants the learner credit that is recorded on the delivery platform for successfully completed attempts. The developer shall configure content in accordance with delivery platform content configuration guidance for all playability testing. Additionally, the developer must perform playability testing that includes pass and fail paths through content where applicable and spans more than one day for the longest learning path in courseware content to verify that the courseware operates correctly in realistic use cases.

See Table 1 on the next page for examples of required testing paths that the developer must follow for an example course’s instructional strategy.
Table 1: Example of Required Instructional Strategy Testing Paths

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Testing Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learner passes phase 1-3 pretests.</td>
<td>Pass the pretest for phase 1 and ensure the learner is taken to the phase 2 pretest. Pass the pretest for phase 2 and ensure the learner is taken to the pretest for phase 3. Pass the pretest for phase 3 and ensure the learner is taken to the capstone exercise. Successfully complete the capstone exercise, then verify learner credit in delivery platform transcript.</td>
</tr>
</tbody>
</table>
| Learner fails each pretest, and passes the first posttest. | Fail the pretest for phase 1 and ensure the learner is taken to the phase 1 lesson. Successfully complete the phase 1 lesson and ensure the learner is taken to the posttest for phase 1. Successfully complete the first phase 1 posttest and ensure the learner is taken to the phase 2 pretest.  
    Repeat the process for phases 2 and 3 until arriving at the capstone exercise. Successfully complete the capstone exercise, then verify learner credit in delivery platform transcript. |
| Learner fails each phase pretest, each phase posttest, remediates, and passes each phase second posttest. | Fail the pretest for phase 1 and ensure the learner is taken to the phase 1 lesson. Successfully complete the phase 1 lesson and ensure the learner is taken to the first posttest for phase 1. Fail the phase 1 posttest and ensure the learner is remediates back to the phase 1 lesson. Successfully complete the phase 1 lesson, then successfully complete the second phase 1 posttest and ensure the learner is taken to the phase 2 pretest.  
    Repeat the process for phases 2 and 3 until arriving at the capstone exercise. Successfully complete the capstone exercise, then verify learner credit in delivery platform transcript. |
| Learner fails phase 1 pretest and posttest, remediates, and fails phase 1 second posttest. | Fail the pretest for phase 1 and ensure the learner is taken to the phase 1 lesson. Successfully complete the phase 1 lesson and ensure the learner is taken to the first phase 1 posttest. Fail the first phase 1 posttest and ensure the learner is remediates back to the phase 1 lesson. Fail the second phase 1 second posttest and verify that the learner is not allowed to progress further and receives an unsuccessful completion mark in delivery platform transcript.  
    Apply the same process to phases 2 and 3 with separate registrations, passing the prior phases to get to them. |
3.5 Delivery Platform Courseware Configuration
The developer shall follow the guidance and documentation provided for loading and configuring courseware content on a delivery platform. Different delivery platforms will require different procedures for setting up content.

For example, courseware that is intended for the ALMS (Saba LCMS platform) has specific processes for setup defined in a variety of documents found on the ALMS’ knowledge base. The ALMS content configuration documentation covers topics such as uploading and configuring SCORM content packages, configuring course constructs with sequencing, and more. An overview of the available documentation is available at: https://www.atsc.army.mil/tadlp/delivery/alms.asp.

Note that other LMSs such as the ELLC (Blackboard LMS platform) may require the developer to contact the administrator for the individual ELLC platform for instructions on loading and setting up content.

3.6 BRBP Checklist for SCORM 2004 CMI
The developer shall complete and submit the Army BRBP Checklist for SCORM 2004 CMI along with conformance testing log files. A detailed explanation of the Army business rules for SCORM courseware and instructions for completing the BRBP checklist are available for viewing and download on the BRBP website at: https://tadlp.github.io/brbp/.

3.7 Critical Errors
Developers are required to have knowledge of the types of errors that may be encountered during the evaluation of Army courseware, and how to determine their criticality rating that affects the decision to field courseware or not. The courseware testing criticality matrix document is available at: https://www.atsc.army.mil/tadlp/implementation/compliance/acceptance_criteria.asp.

The criticality matrix provides standardized information to aid courseware developers, Army activities, and testers in the determination of criticality ratings for errors encountered during test and evaluation. Note that the document is not an all-inclusive list of errors and ratings and is updated as necessary.
3.8 Section 508 Compliance Verification
The developer shall ensure that courseware meets the standards specified in Section 508 of the Rehabilitation Act, Title 29 – Labor, Chapter 16 – Vocational Rehabilitation and Other Rehabilitation Services, Section 794d – Electronic and Information Technology. Section 508 law requires all federal agencies to provide access to electronic and information technologies (EIT) for disabled individuals. The law states:

“Agencies must provide employees and members of the public who have disabilities access to electronic and information technology that is comparable to the access available to employees and members of the public who are not individuals with disabilities.”

To ensure that courseware meets Section 508 standards, the developer must perform the following:

- Design content to meet all standards for software applications, operating systems, and web-based intranet and Internet information and applications as stated in TRADOC Pam. 350-70-12.
- Test and validate content using Section 508 validation and reporting software and methodologies.
- Provide written documentation identifying which accessibility checkpoints that content meets.
- Package and include Section 508 test logs with the content submission package.

4 Government Review Process
This section describes the review process that the Government follows during the development lifecycle of courseware.

4.1 IMDP Review
The Government will review the IMDP for acceptance. IMDP review includes acceptance testing of the content wireframe submitted with the IMDP. After the Government approves the IMDP materials and wireframe, the developer may proceed with the development and assembly of courseware content.

4.2 Incremental Module Testing
The Government will conduct compliance review and subject matter expert (SME) reviews on submitted courseware at the learning event level specified in the Delivery Order and for accompanying learner performance measurement instruments and checks (e.g. checks-on-learning, practical exercises, simulations, and exams). The Government will perform incremental technical reviews as content is developed until the content has no critical technical conformance faults and the Army activity considers the courseware adequate for validation and limited incremental fielding. The developer shall correct all critical technical conformance faults and apply lessons learned from their own testing of content submission to subsequent development. In addition, the developer shall correct any deficiencies identified during validation activates prior to final submission of courseware.
4.3 Final Courseware Approval
The developer shall submit finalized courseware to the Government for approval only after successful validation of content through the incremental testing and review process, through which the developer has corrected all deficiencies identified. Based on courseware usage indicated in the Delivery Order, the developer shall provide appropriate versions of the courseware (e.g. web-based and CD-ROM versions) to the Government. The Government will conduct acceptance testing reviews on the final courseware and supporting documentation (e.g. conformance testing logs, completed BRBP checklists, student transcript screenshots, lesson flow diagrams, etc.) to include all lessons, learning events, learner performance measurement instruments, practical exercises, and tests. The final accepted courseware will replace the incrementally delivered content and be given separate unique version identifier(s).
Appendix A – Workstation Configuration

Before testing can begin, the testing computer workstation(s) specifications should meet, at a minimum, those in the Army Baseline Home Computer Configuration (BHCC) for Army Distributed Learning Courseware for FY18+.

The BHCC for FY18+ can be accessed at: https://www.atsc.army.mil/tadlp/implementation/config/home_computer.asp.

Additionally, testing requires the following software to be installed on the workstation(s):

- Latest version of the 32-bit Java Runtime Environment (JRE)
- ADL SCORM 2004 3rd Edition Conformance Test Suite v1.1.2
- SCORM Resource Validator

It is recommended that the developer create a dedicated folder in which to store the log files generated by conformance testing tools, and have text editing software available.
Appendix B – Understanding the ADL Test Suite

This appendix provides information on the ADL SCORM 2004 3rd Edition Conformance Test Suite and the functions it performs. Before testing SCORM courseware, it is critical that the developer (referred to as the tester in this appendix) has an understanding of the SCORM 2004 3rd Edition specification and its implementation; the official documentation for the specification is a useful resource:

- SCORM 2004 3rd Edition technical documentation


NOTE: The latest version of the ADL SCORM 2004 3rd Edition Conformance Test Suite is version 1.1.2. The tester must ensure they are using the correct version of the test suite.

Conformance Testing Overview

Upon launching the ADL SCORM 2004 3rd Edition Conformance Test Suite, note the test options shown:

- **Content Package Conformance Test**: The primary test that the tester will run to test overall conformance of SCORM 2004 3rd Edition courseware packages. The content package conformance test combines the Sharable Content Object (SCO) Run-Time Environment (RTE) and manifest utility tests into one workflow and produces the log files required by Army acceptance criteria.

- **SCO RTE Conformance Utility Test**: Tests SCO run-time performance for conformance to the SCORM 2004 3rd Edition specification without testing the conformance of the packaging around it (i.e., the package manifest). The test checks for minimum communication activity via the SCORM API, and validates all communication occurs during run-time against the SCORM 2004 3rd Edition specification.

- **Manifest Utility Test**: Tests the manifest file for a SCORM 2004 3rd Edition content package without testing the SCO(s) for run-time performance. The test checks a package’s manifest file for well-formedness, validity against the controlling document, ADL mandatory rules and extensions, and valid references for all resources.

In most cases, the tester will run choose the **Content Package Conformance Test** option. The SCO RTE and manifest utility tests can be run individually for diagnostic testing, but the tester must run the full content package conformance test to verify conformance of SCORM 2004 3rd edition courseware.
Content Package Conformance Test Workflow
This section contains a walkthrough of the ADL SCORM 2004 3rd Edition Conformance Test Suite’s Content Package Conformance Test workflow.

Opening the Test Suite
When the tester opens the test suite in their web browser, they will see the initial screen that lists the available tests described in the overview section of this appendix. The tester selects the Content Package Conformance Test to produce the log files required by Army acceptance criteria.

Figure B-1: Initial screen of the ADL test suite

Loading the Content Package Conformance Test
After selecting the option for the Content Package Conformance Test on the main screen, the tester must wait a short time for two security prompts to appear asking them to allow the ADL test suite Java applets to run. To allow the test to run, the tester selects the “Allow” button.

Figure B-2: Internet Explorer security prompt for test suite Java applet
Once the test suite applets have loaded correctly, informational text will appear on the right side of the screen. If the name of the test suite, its version, and the “Self Test Log” heading do not appear, the Java applets failed to load. If the test suite Java applets do not load, the tester must close the browser window and verify that the testing machine’s Java and web browser configuration settings allow applets to run through the web browser’s java plugin.

![Figure B-3: Test suite information shown after Java applets load](image)

**Configuring the Content Package Conformance Test**

After the test suite has fully loaded, the tester follows the instructions presented and enters information to identify the SCORM content package being tested:

- Content Package Name
- Content Package Version/Release Number
- Content Package Vendor/Developer

![Figure B-4: Sample content package identifying information](image)
The tester then indicates the type of content package test to run:

- **Content Package (PIF):** Test a content package file (PKZIP 2.04g format; .zip).
- **Content Package (non-PIF):** Test the contents of a package on the local file system, media disk, or network file system. The tester must extract all files from a content package and enter the manifest file in step 4 (instead of content package) when selecting this option.

The “Content Package (PIF)” option is recommended, since a content package PIF is the deliverable item loaded on the delivery platform.

The tester then indicates the type of SCORM Application Profile the content to be tested uses:

- **Content Aggregation Content Package:** A content package that contains one or more organizations.
- **Resource Content Package:** A content package that contains no organizational structure, but instead is a list of resources.

The “Content Aggregation Content Package” option must be selected, since Army SCORM courseware will always have an organizational structure.
The tester then indicates the location of the content package to be tested and selects the “Begin Test” button.

![Select the Content Package to be tested](image)

Figure B-7: Selecting a content package for testing

The test suite will then validate the conformance of the content package’s manifest and overall conformance to the SCORM content aggregation model (CAM) part of the SCORM 2004 3rd Edition specification. The results of the tests will appear on the right side of the screen.

![The test suite after successfully testing CAM conformance of a package](image)

Figure B-8: The test suite after successfully testing CAM conformance of a package

If the tester wishes to see the detailed results of the content package test in the event of an error, or for confirmation, they can select the “Click here to see the detailed CP test log” link shown on the right side of the test suite.

**Configuring the Content Package Conformance Test SCO RTE Portion**

After the content package test completes successfully, the tester will configure the SCO RTE portion to test the SCO(s) in the package and validate their behavior against the SCORM 2004 3rd Edition specification.
The tester enters a timeout value that indicates how long the test suite should wait for the content to call the SCORM 2004 API “Initialize” method before automatically failing the test. The default value is 20 seconds, and it is recommended to leave the value as is.

The tester then has the option to enter data that the delivery platform would provide to the SCO(s):

- Learner ID
- Learner Name
- Credit or No Credit
- Mode

It is recommended that the tester not enter any data, as the test suite will provide default values. Army acceptance criteria does not require the tester to manually enter delivery platform data in this portion of the test.

The tester then has the option to launch the SCO(s) or abort the test and start over. Selecting the “Launch SCO(s)” button will launch the first SCO detected in the content package in a new browser window.
In the new browser window, the SCO content will be shown in the upper majority of the screen, and a “SCO Run Time Environment Test” banner is shown on the bottom portion. The tester navigates through the content according to their desired testing path and the instructional strategy, and exits the content as the learner would normally.

**WARNING:** The SCO RTE test does not provide delivery platform controls such as an LMS “Back” and “Forward” buttons. If content relies on delivery platform navigation controls, the tester may not be able to fully test their content.

**NOTE:** Depending on the configuration of the content, the new browser window might not close when the tester selects whichever button is programmed to exit the content. This is normal, and after selecting the button to exit the content, the tester should return to the main test suite browser window.

After completing the SCO, the tester returns to the main test suite browser window and selects the “Complete Test” button. If there are additional SCOs, the test suite will then launch them, and the process is repeated until all SCOs are tested.
Reviewing Conformance Results

The test suite shows a summary of the SCO RTE conformance test results on the right side of the screen.

If the tester wishes to see the detailed results of the content package test in the event of an error, or for confirmation, they can select the “Click here to view the detailed SCO test log” link shown on the right side of the test suite.

When all tests are complete, the bottom right portion of the screen will display conformance statements for the package as a whole, the content aggregation model (CAM) conformance tested during the content package portion of the test, and the run-time environment (RTE) tested during the second portion.

If any part of the testing produced non-conformant results, the conformance statements will state which part of the test failed, and the tester will have to investigate the issue(s), remediate them, and fully retest the content.

Figure B13: The test suite after successfully testing SCO RTE conformance of a package

Figure B14: Example conformance statements for a non-conformant package
If all tests completed successfully, the test suite while show a message indicating that the content package is conformant to the SCORM 2004 3rd edition specification.

![SCORM 2004 3rd Edition Conformance Statement](image)

**Figure B-15: Example conformance statements for a conformant package**

The Content Package Conformance Test will produce the following log files regardless of the results of the various tests performed:

- **Content Package Summary Log**: An XML log file that contains a summary of conformance results from individual tests (content package, metadata and run-time environment).
- **Content Package Detailed Log**: An XML log file that contains detailed results of conformance tests performed on the content package itself based on the package manifest’s construction.
- **Metadata Detailed Log**: An XML log file for each metadata file detected in the package manifest that contains detailed results of conformance tests performed on the metadata file.
- **SCO Detailed Log**: An XML log for each SCO defined in the content package manifest that contains detailed results of conformance tests performed on the run-time behavior and communication through the SCORM API.

Each log will contain the date and time of the test log, and the name of the package or SCO where applicable, as illustrated in the figure below.

<table>
<thead>
<tr>
<th>ADL SCORM 2004 3rd Edition Conformance Test Suite Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Name</td>
</tr>
<tr>
<td>Content Package Summary Log</td>
</tr>
<tr>
<td>Content Package Detailed Log</td>
</tr>
<tr>
<td>Metadata Detailed Log</td>
</tr>
<tr>
<td>SCO Detailed Log</td>
</tr>
</tbody>
</table>

**Figure B-16: ADL SCORM 2004 3rd Edition Conformance Test Suite log files**

Each log file can be opened in either a text editor or in Internet Explorer for a more visually-friendly view, provided the “util” folder is in the same folder as the log files.

ADL test suite log files have a default location of the “Logs” folder in the test suite directory, with a separate folder for each type of test. For example, Content Package Conformance Test logs can exist in a new folder named with the test date, time, and package name in the “Logs” folder at:

- C:\ADL\SCORM_2004_3rd_Ed_CTS_V1.1.2_ST\TestSuite\Logs\2018-05-09_09.56.36541-Test
The content package summary log file contains a summary listing of the tests performed on the manifest, metadata, and SCOs, as shown in Figure B-17.

![Figure B-17: Example ADL test suite content package summary log](image)

---

**Figure B-17: Example ADL test suite content package summary log**
The summary overview is useful for identifying areas of non-conformance, and the log provides links to the detailed logs in the set that contain individual tests items. Each detailed log (content package, metadata, and SCO) show every test item, and the result of each test. Figure B-18 illustrates a portion of the SCO detailed log showing some of the SCORM API communication that took place during the test.

<table>
<thead>
<tr>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempting to Launch SCO: C:\ADL\SCORM_2004_3rd_Ed_CTS_V1.1.2 ST\PackageImport\content\3207 \launch.html</td>
</tr>
<tr>
<td>The SCO is searching for the API Instance ...</td>
</tr>
<tr>
<td>✓ The SCO was able to find the API instance</td>
</tr>
<tr>
<td>✓ Initialize(&quot;&quot;) has been called</td>
</tr>
<tr>
<td>✓ The Initialize() method finished successfully</td>
</tr>
<tr>
<td>✓ The GetLastError() method has been called</td>
</tr>
<tr>
<td>✓ The Last Error returned: 0</td>
</tr>
<tr>
<td>✓ The GetLastError() method call finished successfully</td>
</tr>
<tr>
<td>✓ GetValue(&quot;cmi.completion_status&quot;) has been called</td>
</tr>
<tr>
<td>✓ The value returned from the GetValue() method call: [ unknown ]</td>
</tr>
<tr>
<td>✓ The GetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ The GetLastError() method has been called</td>
</tr>
<tr>
<td>✓ The Last Error returned: 0</td>
</tr>
<tr>
<td>✓ The GetLastError() method call finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.completion_status&quot;,&quot;incomplete&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.suspend_data&quot;,&quot;{&quot;children&quot;:{&quot;C&quot;:1},{}},{},{}{},{}{},{}{},{}{}&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.location&quot;,&quot;1&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.suspend_data&quot;,&quot;{&quot;children&quot;:{&quot;C&quot;:1},{}},{},{}{},{}{},{}{},{}{}&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.location&quot;,&quot;2&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ Commit(&quot;&quot;) has been called</td>
</tr>
<tr>
<td>✓ The Commit() method call finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.suspend_data&quot;,&quot;{&quot;children&quot;:{&quot;C&quot;:1},{&quot;C&quot;:1},{}{},{}{},{}{},{}{},{}{},{}{},{}{}&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ SetValue(&quot;cmi.location&quot;,&quot;3&quot;) has been called</td>
</tr>
<tr>
<td>✓ The SetValue() method finished successfully</td>
</tr>
<tr>
<td>✓ Commit(&quot;&quot;&quot;) has been called</td>
</tr>
<tr>
<td>✓ The Commit() method call finished successfully</td>
</tr>
</tbody>
</table>

Figure B-18: Example ADL test suite SCO detailed log

Detailed logs are critical in helping the developer identify the root cause of a non-conformant rating because of the granularity they provide.

Once the tester has reviewed all the logs from the ADL test suite and has a fully conformant set, they may then proceed to performing additional Army conformance tests using the SCORM Resource Validator.
Appendix C – The SCORM Resource Validator

This appendix provides information on using the SCORM Resource Validator and the functions it performs. The SCORM Resource Validator inspects a SCORM content package (a “package interchange file” or “PIF”) to determine any compliance issues that may exist and to facilitate the SCORM testing workflow described in this appendix.

NOTE: The latest version of the SCORM Resource Validator is version 2. The tester must ensure they are using the correct version of the application.

SCORM Resource Validator Overview

The SCORM Resource Validator performs the following actions to conduct its conformance testing:

1. Uses the "imsmanifest.xml" file from the PIF to locate the metadata files in the package. Once located, the contents of the metadata files are compared to the SCORM allowable values and the Army-required values for conformance.

2. Compares the physical files found in the PIF to those referenced in the manifest to verify if the files listed on the manifest exist, as well as verifying that files contained in the PIF are listed in the manifest.

3. If a set of ADL Test Suite log files (see Appendix B) are included, the application will determine if the logs contain Army-required SCORM Runtime API calls. The Resource Validator will also create a summarized log of all the required conformance information from the ADL test suite logs and the information gathered by the Resource Validator.

Metadata conformance

The Army requires that all metadata files be external to the manifest in a separate referenced XML file. The Army also requires that certain metadata elements to be present on all metadata files submitted with the courseware in addition to the SCORM-required metadata.

The Resource Validator validates that mandatory Army metadata elements exist in metadata files. If the metadata files are not listed in the "imsmanifest.xml" file, this violates the requirement of full disclosure of all files in the content package. Any metadata files that have been excluded from the manifest file will not be checked.

Reusability Conformance

The SCORM specification requires that all files pertaining to a SCO be listed on the manifest file in the <file> tags contained within the <resource> tag. This ensures the "reusability" designated by the SCORM 2004 3rd Edition specification.

The problem may not be apparent to the tester because even if some files are omitted, the SCO will probably play normally in the delivery platform. The problem occurs when a SCO is extracted from a courseware content package and the SCO is reused in another course.

For a SCO to be reused in another course, one SCO is selected from an existing manifest file, and only those files that are listed on the manifest file associated with that SCO will be extracted to the new package. If all the files are not listed on the manifest, then an incomplete SCO will be transferred.
The SCORM Resource Validator performs the following two tests on the input PIF:

- Manifest File Validation
- Package File Validation

**Manifest File Validation**

The SCORM Resource Validator verifies if the files listed on the manifest actually exist in the courseware content package.

<table>
<thead>
<tr>
<th>Files as listed on the manifest</th>
<th>Files existing in the directories</th>
</tr>
</thead>
</table>
| `<resource identifier="SCO1" type="webcontent" adlcp:scormtype="sco" href="safety/unit/index.html"> <metadata>... </metadata>  
  <file href="safety/unit/index.html"/>  
  <file href="safety/unit/page2.html"/>  
  <file href="safety/unit/page3.html"/>  
  <file href="safety/unit/page4.html"/>  
</resource>` | ![safety]  
  ![unit]  
  ![index.html]  
  ![page2.html]  
  ![page4.html] |

**Result:**

"page3.html" is listed on the manifest but does not exist in the directories. This manifest is inaccurate.

**Package File Validation**

The SCORM Resource Validator verifies files that are contained in the courseware content package are listed on the manifest. This test will determine whether the manifest is a "complete" or "incomplete" manifest. An "incomplete" manifest is a manifest in which all files contained in the courseware content package are not listed on the manifest.

<table>
<thead>
<tr>
<th>Files as listed on the manifest</th>
<th>Files existing in the directories</th>
</tr>
</thead>
</table>
| `<resource identifier="SCO1" type="webcontent" adlcp:scormtype="sco" href="safety/unit/index.html"> <metadata>... </metadata>  
  <file href="safety/unit/index.html"/>  
  <file href="safety/unit/page2.html"/>  
  <file href="safety/unit/page3.html"/>  
  <file href="safety/unit/page4.html"/>  
  <file href="safety/unit/page5.html"/>  
  <file href="safety/unit/page6.html"/>  
  <file href="safety/unit/graphic.jpg"/>  
</resource>` | ![safety]  
  ![unit]  
  ![index.html]  
  ![page2.html]  
  ![page3.html]  
  ![page4.html]  
  ![page5.html]  
  ![page6.html]  
  ![graphic.jpg] |

**Result:**

There are 3 files ("page5.html", "page6.html" and "graphic.jpg") are in the PIF directories, but these three files are not listed on the manifest. This manifest is incomplete.
SCORM Resource Validator Workflow
This section contains a walkthrough of the SCORM Resource Validator’s testing workflow.

Configuring the SCORM Resource Validator
When the tester opens the resource validator, they must input a SCORM content package to test.

Figure C-1: SCORM Resource Validator after launch
The tester selects the “Browse” button to open a file browser dialog.

![Image](image1.png)

**Figure C-2: The “Browse” button in the SCORM Resource Validator**

The tester selects the SCORM content package file supplied for testing using the file browser dialog and selects the “Open” button.

![Image](image2.png)

**Figure C-3: The file browser dialog open after selecting the “Browse” button**
The tester then has the option of providing conformance test logs from the ADL SCORM 2004 3rd Edition Conformance Test Suite. It is recommended that the tester has ADL test suite log files available from prior testing of the content package.

The tester selects the “Browse” button that is now active in the section at the bottom of the application under the “Select directory to parse ADL Test Suite log files” heading.

Figure C-4: The SCORM Resource Validator after selecting a PIF
The tester uses the folder browser dialog to select the folder where the previously generated ADL Test suite log files supplied for review are located and selects the “OK” button.

![Folder browser dialog](image)

**Figure C-5:** The folder browser dialog shown after selecting the “Browse” button

The tester then selects the “Validate” button in the upper right corner of the resource validator.

![Validate button](image)

**Figure C-6:** The “Validate” button shown in the upper right of the application.
Reviewing Conformance Results

The Resource Validator conducts its conformance tests and displays the results in the four text areas in the main body of the application. A message shown at the lower right corner of the application indicates the result of tests run on the ADL test suite logs supplied.

The tester can select the “View Log Files” link to open the folder that contains the log files generated by the SCORM Resource Validator to review the logs.

Figure C-7: The SCORM Resource Validator after testing a PIF and ADL test suite logs
The SCORM Resource Validator will produce the following log files:

- **Army Run-Time Validation Log**: An HTML file that contains the results of tests performed by parsing the content of ADL test suite log files for Army requirements.

- **Army Metadata Validation Log**: A DOC file that contains the results of conformance tests run on metadata files detected in the package manifest.

- **Content Package Resource Validation Logs**: A set of four HTML files that contain the results of conformance tests for content package resources:
  - **Manifest Resource Log**: A list of files referenced in the content package manifest as resources.
  - **Content Package Resource Log**: A list of files detected within the content package itself.
  - **Manifest-to-Content-Package Resource Log**: A list of files found in the content package manifest but not found in the content package itself.
  - **Content-Package-to-Manifest-Resource Log**: A list of files found in the content package itself but not referenced in the manifest file.

The SCORM Resource Validator log files are named as shown in the figure below.

<table>
<thead>
<tr>
<th>SCORM Resource Validator Logs</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Army Run-Time Validation Log</td>
<td>ParsedOutput.html</td>
</tr>
<tr>
<td>Army Metadata Validation Log</td>
<td>ValidateMD.doc</td>
</tr>
<tr>
<td>Manifest Resource Log</td>
<td>manifest_files_found.html</td>
</tr>
<tr>
<td>Content Package Resource Log</td>
<td>packaged_files_found.html</td>
</tr>
<tr>
<td>Manifest-to-Content-Package Resource Log</td>
<td>manifest_files_missing.html</td>
</tr>
<tr>
<td>Content-Package-to-Manifest-Resource Log</td>
<td>packaged_files_missing.html</td>
</tr>
</tbody>
</table>

**Figure C-8: SCORM Resource Validator log files**

Note that there is also an additional file named “PIF_file_validate.txt” that contains a checksum value for the supplied content package that can be used to verify content packet file integrity when it is transferred.

Log files will be placed in a “logs” folder in the same directory as the SCORM Resource Validator. Each package tested will result in a new folder for the package’s logs. For example, a set of logs can exist at:

- `C:\SCORM_RV\logs\RV_180801_041159t1-1`