



**Aspen Learning Management Server**

---

**SCORM-ready, Aspen-ready**  
**A Guide to Making Content Compatible with**  
**Aspen LMS Versions 1.1 and 2.0**

click **2**learn

---

# Contents

<b>Introduction.....</b>	<b>3</b>
<b>About SCORM .....</b>	<b>3</b>
<b>What is SCORM-compliant Learning Content? .....</b>	<b>3</b>
<b>Creating a SCORM Package .....</b>	<b>3</b>
<b>Testing the SCORM Package for Conformance.....</b>	<b>4</b>
Working with the SCORM 1.2 Test Suite .....	5
About the SCORM 1.2 Sample Run-time Environment.....	5
<b>If Your Package Fails Conformance .....</b>	<b>5</b>
<b>Considerations Specific to Aspen .....</b>	<b>6</b>
Packaging Considerations .....	6
Run-time Behaviors .....	7

SCORM-ready, Aspen-ready

Copyright © 2002 Click2learn, Inc. All rights reserved. Click2learn, the Click2learn logo, Aspen, the Aspen logo, Aspen Learning Management Server, Aspen Learning Content Management Server, and Aspen Virtual Classroom Server are trademarks of Click2learn, Inc. All other company and/or product names are the property of their respective owners.

Information in this document is subject to change without notice.



## Introduction

Prior to publishing in Aspen, it is critical to verify that your SCORM 1.2 compliant content can be easily published and delivered by the Aspen Learning Management Server (LMS). This document applies to Aspen versions 1.1 and 2.0.

This document provides you with the specific steps you will need to take in order to verify compliance. It also provides some best practice guidelines that may not be clear from reading the SCORM specification. The verification process can be done entirely offline; an LMS is not required. To use this document, you should have basic familiarity with Web concepts, computer setup and basic HTML.

## About SCORM

SCORM is a set of specifications that describes:

- How to create Web-based learning content that can be delivered and tracked by different SCORM-compliant learning management systems.
- What a SCORM-compliant learning management system must do in order to properly deliver and track SCORM-compliant learning content.
- The SCORM specifications, which are based on various other industry standards and specification. (The current official version is 1.2.)

The SCORM specification does not cover all aspects of a learning enterprise. For example, it does not specify how tracking information is stored, what reports are generated, what pedagogical or training models should be used, or how learner information is compiled.

## What is SCORM-compliant Learning Content?

In the SCORM 1.2 specification, SCORM-compliant learning content is:

- Delivered through a Web browser.
- Described by metadata.
- Organized as a collection or sequence.
- Packaged with a manifest file in such a way that it can be imported by a compliant learning management system or into a repository used by such a system.

SCORM content is made of learning objects aggregated into a content package. Learning objects are units of content that can be delivered to a learner by a SCORM-compliant learning management system in order to create a useful learning experience.

*Note:* The learning objects used in a SCORM-compliant package may be fully included in the package or used by reference. However, because the behaviors and browser security issues associated with running content from a foreign server are not well addressed by standards at this time, SCORM content for delivery by Aspen should be fully included in the package.

## Creating a SCORM Package

Each SCORM package can be published as a course (or "learning experience") in Aspen. You can build a SCORM package by hand or by using a more or less automated tool or authoring system such as Aspen LCMS or ToolBook. However you create your SCORM content package, you must end up with the following:

- A set of files that can all reside in the same main directory, and/or in a directory tree of which that main directory is the root.
- One of the files in the main directory must be a manifest file with the name `imsmanifest.xml`. This file must reside in the main directory.
- The package must include metadata. The metadata may be embedded inline within the `imsmanifest.xml` file, or it may be in a separate xml file that is referenced by the `imsmanifest.xml` file.
- None of the files may contain references or links to any file outside the tree.
- None of the files may contain any reference or link that specifies a hard-wired path. Relative paths are acceptable. Hard wired paths that contain a drive letter or that assume that the tree will end up in any particular place in a file system are not acceptable. No path in a reference or URL may begin with "/" or "\".
- HTML content may not assume that it will run in a top-level window, or attempt to force itself to run into a top-level window. It must be designed so that it can be launched in a frame.
- A Sharable Content Object (SCO) may open a dependent popup window but this not considered a best practice. In any case, the SCO is responsible for closing such a window before the SCO exits.
- A SCO may not contain a hyperlink to another SCO. A SCO may not assume that any other SCO will be present in the system.
- SCORM 1.2 does not define any navigation behavior, and you should not assume that one is provided. The only defined behavior is that the user will be given a way to choose which SCO to run. In particular, one should not assume that calling `LMSFinish` will automatically result in continuing to the next SCO.
- A SCO may consist of more than one HTML page. However it is recommended that SCOs that contain multiple pages use a mechanism to maintain state across pages. This includes delivering them through a frameset, a "slave" window or cookies. Note that some enterprises do not allow cookies. For an example of using a frameset for multiple pages, see *Cooking Up a SCORM – Version 1.2* at <http://home.click2learn.com/standardswork>.
- Any `.xsd` file referenced in the manifest or metadata file must be included in the package.
- The standard `.xsd` files may not be modified in any way. If your package manifest or metadata require extensions, those must use one or more separate namespaces. The namespaces must be properly defined in `.xsd` files included in the package.
- The set of files may be collected into a `.zip` archive. However, this is not a requirement.

If you need specific help in constructing your package, it is likely that your questions have already been asked and answered in the ADL forums at <http://www.adlnet.org>. Registration is free and, while the quality of responses varies widely, experts monitor these forums.

## Testing the SCORM Package for Conformance

If you have any doubt about whether the content package will work within the Aspen Suite, you should determine whether it is conformant with SCORM standards.

### ► To verify that your package conforms with SCORM 1.2:

- 1** Review the *SCORM Version 1.2 Conformance Requirements Version 1.2* document, which can be downloaded from the Resource Center at <http://www.adlnet.org>.
- 2** Run your package through the SCORM Version 1.2 Conformance Test Suite Version 1.2.3 (Self Test), which can be downloaded from the Resource Center at <http://www.adlnet.org>. (See the specific steps outlined in the next section.)
- 3** Run the package through a test run-time environment and look for unexpected behaviors or errors. You can download a SCORM Version 1.2 Sample Run-Time Environment from the ADL Web site.

Note that this is only a sample run-time environment. It does not necessarily behave like other run-time environments in the way it handles features that are not specifically defined in the SCORM specification.

## Working with the SCORM 1.2 Test Suite

To use the SCORM 1.2 test suite (available at <http://www.adlnet.org>), follow the directions below.

### ► To install the test suite:

- a** Read the readme.htm file before you begin the installation. It contains critical information about the particular version of the Java runtime environment required by the test suite. If you do not use exactly that version, you will have to uninstall and reinstall the test suite before it works correctly. Note that the version required is *not* the most recent version of the Java runtime.
- b** Install the test suite.
- c** If the test suite is already installed, check the ADL web site for a new version. As of this writing the current version is 1.2.3. Be sure you use the current version and not an outdated version.
- d** The manifest file in your package includes references to various .xsd files that should be included in the package. Copy those files to the desktop on the test machine. The test suite cannot find them unless they are there, and it does need to reference them.
- e** If your manifest or metadata include extensions, copy the namespace .xsd files for those extensions to the desktop as well. These must be the same .xsd files you include in the package.

### ► To launch the test suite:

- a** From the Start menu, select **Programs**, then **ADL**. Choose **Test Suite 1.2.3**, then **SCORM™**.
- b** From the menu, select **Content Package Conformance Test**.
- c** In the Package test option, choose **Package (non PIF)** or **Package (PIF)** depending on whether the generated package was zipped.
- d** In the window that appears, choose **Content Aggregation Package**.
- e** In the window that appears, click **Browse** and locate either the .zip file or the imsmanifest.xml file for the package.
- f** Complete all the test suite steps. Whenever prompted, click **Continue Test**.

It is recommended that you save the test log, and also retrieve the detail log from the test suite temporary directory. Saving these logs will provide Click2learn's Product Support team or a SCORM consultant diagnose and solve content issues. Typically, this test log is a file named C:\Program Files\ADL\TestSuite1\_2\_3\contentpackage\tempdir\Log\_XXXXXXX\CP\_log.htm, where XXXXXXXX is the date and time of the test.

## About the SCORM 1.2 Sample Run-time Environment

The ADL sample run-time environment emulates a single-user Learning Management System. Refer to the documentation that comes with the Run-Time Environment to install and run it. The documentation includes a link to the ADL help desk.

## If Your Package Fails Conformance

If your package fails the SCORM test suite, it will almost certainly not run in Aspen. As a best practice, do not invest time and resources trying to upload it to Aspen until the package passes the test suite.

Note also that the test suite may occasionally be at fault and signal errors when there are none. This may occur due to a configuration error or a browser error. For example, on at least one Windows 2000 system

with Internet Explorer, it has been observed that the test suite “hijacks” any open Windows Explorer window and launches into the right pane. This is guaranteed to fail. To verify that the test suite is not at fault, try to test a package known to be “good” and verify that you follow the test procedure as documented. “Good” packages can be downloaded from the ADL Web site.

► **To ensure your package is conformant:**

- 1** Review the error logs against the SCORM Version 1.2 Conformance Requirements Version 1.2 document.
- 2** Check that there is no typo in the imsmanifest.xml file, particularly in the section that contains the namespace declarations. If necessary, copy and paste that part of the manifest from a "known good" file.
- 3** Go to the IMS Global Consortium web site at [www.imsglobal.org](http://www.imsglobal.org) and download the specifications for Content Packaging and Metadata. The SCORM packaging and metadata specifications are an application profile of those specifications. Those original documents, especially the Best Practice Guides, contain explanations that may help you in interpreting SCORM.
- 4** Make sure you use the version that is specified in SCORM 1.2 for each of the XML schema files (.xsd files). These files are available on the IMS Web site, as well as in the SCORM package sample that comes with the test suite. These are not necessarily the latest version of the .xsd files posted on the IMS Web site. Some versions posted on the IMS Web site have not undergone the extensive testing of the versions referenced in SCORM 1.2 review and were released in answer to some specific issues that do not affect SCORM 1.2 content.
- 5** If your content was created with a publishing template, such as one of the Macromedia extensions, verify that you have the most current version of the template.
- 6** Check the Click2learn support Web page and the <http://home.click2learn.com/standardswork> unsupported Web page. As they become available, utilities that can help you debug your SCORM content will be posted there.

## Considerations Specific to Aspen

There are some considerations specific to packaging and run-time behaviors in the Aspen environment.

### Packaging Considerations

- **Aspen 2.0:** To publish a SCORM package in Aspen 2.0, your package must be in a zip file. Aspen will take the metadata it can use from the package. You can edit or fill in missing metadata as part of the publishing process. Aspen uses a subset of the metadata allowed by the SCORM and IMS metadata specifications. The entire metadata instance from the package is stored in the Aspen database, even though Aspen itself does not use or provide a user interface to all the metadata that may be in that instance.
- **Aspen 1.1:** To publish a SCORM package in Aspen 1.1, it must be available as files in a file system; in other words, if your package is zipped you must unzip it before publishing in Aspen 1.1. Aspen 1.1 does not extract the metadata it can use from the package. You must complete the missing metadata as part of the publishing process. Aspen uses a subset of the metadata allowed by the SCORM and IMS metadata specifications. The entire metadata instance from the package is stored in the Aspen database, even though Aspen itself does not use or provide a user interface to the metadata that may be in that instance.

## Run-time Behaviors

Since SCORM 1.2 does not define behaviors, Aspen manages tracking data in predictable ways based on best practices.

- **Roll up of tracking data:** There are three considerations when rolling up tracking data.
  - a Aspen rolls up course score based on a non-weighted average of all SCOs that report a score.
  - b Aspen rolls up completion based on completion of all SCOs. For the purpose of this rollup, completion is also assumed when the status reported for the SCO is "passed" or "failed".
  - c Aspen rolls up session durations for the all the SCOs. Where a SCO does not report a session\_time value, Aspen provides a session\_time based on an internal timer.
  
- **Stage window:** Aspen launches SCOs in a frame. The dimensions of this stage window are what is left of the actual Aspen window size after display of the table of content on the left (200 pixels) and Aspen management bar on the top (50 pixels). If your package contains only one SCO, no table of content is displayed and the stage window for the SCO is as wide as the Aspen window. It is recommended that your content be tested to be usable in a frame as small as 600x550 if supporting screen resolutions as low as 800x600 is important.

Note that content created with Aspen LCMS can request some Aspen LMS display options through the use of proprietary extensions that are not defined in SCORM 1.2 and thus not honored by systems other than Aspen.