



Learning Management Server

**Working with the API for
SCORM Extensions in Aspen**

Contents

| | |
|--|-----------|
| Introduction | 3 |
| Purpose for SCORM Extensions in Aspen | 3 |
| SCORM Packaging Extensions | 4 |
| SCORM Content Packaging Model | 4 |
| SCORM Packaging Extensions Information Model | 5 |
| Example of Extended Properties | 6 |
| Single-SCO versus Multiple-SCO Content | 7 |
| SCORM Run-time Extensions | 7 |
| SCORM Run-time Extension API Object..... | 7 |
| Aspen Run-time Extensions API..... | 9 |
| SCORM Run-time Extensions API Commands | 10 |
| Modifying the Aspen Content Player | 10 |

Introduction

The SCORM (Shareable Content Object Reference Model) specification defines a standard way for web-based content to communicate with a Learning Management System (LMS). Developed by the Advanced Distributed Learning project (ADL), the SCORM specification defines both a packaging model and a run-time environment. For more information about the SCORM specification, visit www.adlnet.org.

This document is intended for programmers. It explains how to develop SCORM content to take advantage of SCORM extensions available in Aspen 2.5 and Aspen 2.1. This document assumes you are familiar with developing SCORM content and can already develop content that passes the SCORM 1.2 Test Suite for both packaging and run-time. Developing content to interface with the Aspen extended SCORM implementation is not covered by the standard Click2learn technical support programs. Assistance in this area will be provided as a paid services engagement only.

Aspen 2.x is SCORM 1.2 conformant and has been certified level RTE3 conformant by the ADL. The extensions discussed in this document do not violate this conformance. The Aspen SCORM extensions described in this document apply to versions 2.5 and 2.1 of Aspen and SCORM 1.2 content.

Purpose for SCORM Extensions in Aspen

There are some LMS and content considerations that the SCORM specification does not address. In Aspen, the SCORM extensions were created to address these additional areas:

- Allowing the course to define the appropriate size for the run-time player. Aspen defines a default size for content that may be smaller or larger than the size the content specifies. Many content developers use Flash files which define a specific size for Flash content, and the run-time player may not open to this size by default. Because of window size differences the course controls may be hidden, leading to a poor end-user experience. In some cases content will not play if the window is not the proper size.
- Allowing the course to control, to a degree, the appearance of the run-time player. Aspen runs SCORM content in player window that is bordered by graphical elements (the player “footprint”). This footprint may clash with the content or take up valuable screen real estate. Additionally, course navigation is controlled by Aspen through a Table of Contents (TOC) which has a default color and size. We recognize that course designers may wish to hide the player footprint or change the look and feel of the player.
- Allowing the course to navigate from one SCO to another SCO (inter-SCO navigation). The SCORM specification treats individual SCOs (Shareable Content Objects) as autonomous units. A SCO is not aware of its relationship to other SCOs in a course. We recognize that this is a limitation to course designers and Aspen allows navigation forward and backward through the course tree between SCOs.
- Allowing the course to control run-time player functions such as exiting. The Aspen run-time player has additional functions not defined by the SCORM. The exit command, for example, saves all learner data cached in the client side of the Aspen SCORM player before closing the player window. This was added as a player function to provide the course its own exit function without calling `top.close()`. Calling `top.close()` from the content will prematurely close the player resulting in data loss.
- There are additional functions provided by the Aspen run-time player for viewing course progress and study aids. These functions, typically available on the player navigation bar, can be accessed through the Aspen Extensions API.

SCORM Packaging Extensions

The SCORM specification defines a packaging methodology for content whereby all resources in a package are described in the imsmanifest.xml file. This file is described in the SCORM 1.2 Content Aggregation Model document.

SCORM Content Packaging Model

Course elements are described in the imsmanifest.xml file. SCORM content designed to play in Aspen must pass the SCORM 1.2 Packaging Test Suite. Adding Aspen SCORM Extensions to imsmanifest.xml will not make content non-conformant.

The root element for Aspen SCORM Extensions is <c2lcp:ItemDataExtra> and this element should always be a child of <manifest>. This element uses the namespace c2lcp and it must be identified in the <manifest> element. The example below shows the proper placement of <c2lcp:ItemDataExtra> as well as the proper entries to be added to the <manifest> element in order to be conformant with SCORM.

Figure 1 Example of imsmanifest.xml file with Aspen SCORM Extensions highlighted

```
<?xml version="1.0">
<manifest identifier="aspen_extension_manifest" version="1.0"
  xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:c2lcp="http://www.click2learn.com/xsd/c2l_cp_rootv1p1"
  xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2
imscp_rootv1p1p2.xsd
  http://www.imsproject.org/xsd/imscmd_rootv1p2p1 imscmd_rootv1p2p1.xsd
  http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd
  http://www.click2learn.com/xsd/c2l_cp_rootv1p1 c2l_cp_rootv1p1.xsd">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
    <lom>
      <general>
        <title>SCORM - Extended API Test</title>
        <description>This course tests the Aspen SCORM
Extensions</description>
      </general>
    </lom>
  </metadata>
  <organizations default="org1">
    <organization identifier="org1">
      <title>Only Org</title>
      <item identifier="item1" identifierref="res1">
        <title>sco 01</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource
      identifier="res1" type="webcontent" adlcp:scormtype="sco"
href="sco.htm"></resource>
  </resources>
  <c2lcp:ItemDataExtra />
</manifest>
```

SCORM Packaging Extensions Information Model

The following table describes the elements in the SCORM Packaging Extension Model. Any empty property elements will use the default value for that property.

Figure 2 Elements in the SCORM Packaging Extension Model

| Level | Element Name | Description | Multiplicity | Data Type |
|-------|-------------------------|--|--------------|------------------|
| 1 | manifest | SCORM-defined element: refer to the SCORM 1.2 Content Aggregation Model documentation | 1 | container |
| 2 | c2lcp:ItemDataExtra | Element that describes Aspen Extensions the SCORM manifest | 0 or 1 | container |
| 3 | c2lcp:ItemData | Element that describes the course | 1 | container |
| 3.1 | Identifier | Attribute that identifies the course; should match the <manifest> identifier attribute | 1 | ID |
| 3.2 | Type | Attribute that describes the type of Item | 1 | must be "course" |
| 4 | c2lcp:ItemSpecificData | Container element | 1 | container |
| 5 | c2lcp:CourseData | Container element | 1 | container |
| 6 | c2lcp:PackageProperties | Container element that describes all extensible properties | 1 | container |
| 7 | c2lcp:CourseDisplay | Container element that describes extensible display properties | 1 | container |
| 8 | c2lcp:TOC | Container element that describes extensible Table of Contents (TOC) properties | 1 | container |
| 9 | c2lcp:MaxWidth | Element that defines, in pixels, the desired width of the TOC Default: 200 | 1 | integer |
| 9 | c2lcp:ScrollBar | Element that turns on or off the TOC scrollbar Default: yes | 1 | yes/no |
| 9 | c2lcp:BackgroundColor | Element that describes the background color of the TOC Default: ffffff (white) | 1 | hex color value |

| Level | Element Name | Description | Multiplicity | Data Type |
|-------|-------------------|---|--------------|-----------|
| 8 | c2lcp:ShowNavBar | Element that turns on or off the player footprint Default: yes | 1 | yes/no |
| 8 | c2lcp:Launch | Container element that describes extensible launch properties | 1 | container |
| 9 | c2lcp:Width | Element that defines, in pixels, the launch width of the content pane of the player window Default: 800 | 1 | integer |
| 9 | c2lcp:Height | Element that defines, in pixels, the launch height of the content pane of the player window Default: 600 | 1 | integer |
| 9 | c2lcp:AllowResize | Element that defines resize behavior of the player window Default: yes | 1 | yes/no |

Example of Extended Properties

The following example shows all the extensible properties in the SCORM Packaging Extension Model. This course, when shown in Aspen, will open with a viewable content area of 787x577 in a non-resizable window. The Aspen run-time player footprint will be shown and the Table of Contents, if shown, will be 150 pixels wide with a scrollbar and white color.

Figure 3 Example of extensible properties in the SCORM Packaging Extension Model

```

<c2lcp:ItemDataExtra>
  <c2lcp:ItemData Identifier "aspen_extension_manifest" Type="Course">
    <c2lcp:ItemSpecificData>
      <c2lcp:CourseData>
        <c2lcp:PackageProperties>
          <c2lcp:CourseDisplay>
            <c2lcp:TOC>
              <c2lcp:MaxWidth>150</c2lcp:MaxWidth>
              <c2lcp:ScrollBar>yes</c2lcp:ScrollBar>
              <c2lcp:BackgroundColor>ffffff</c2lcp:BackgroundColor>
            </c2lcp:TOC>
            <c2lcp:ShowNavBar>yes</c2lcp:ShowNavBar>
          </c2lcp:CourseDisplay>
          <c2lcp:Launch>
            <c2lcp:Width>787</c2lcp:Width>
            <c2lcp:Height>577</c2lcp:Height>
            <c2lcp:AllowResize>no</c2lcp:AllowResize>
          </c2lcp:Launch>
        </c2lcp:PackageProperties>
      </c2lcp:CourseData>
    </c2lcp:ItemSpecificData>
  </c2lcp:ItemData>
</c2lcp:ItemDataExtra>

```

Single-SCO versus Multiple-SCO Content

Content that has only one item defined in a given organization (one lesson) is referred to as single-SCO content. The example above (figure 1) describes single-SCO content. Multiple-SCO content has multiple items in a given organization.

The Table of Contents is never shown for single-SCO content. The TOC is open by default for multiple-SCO content.

SCORM Run-time Extensions

The SCORM specification defines a run-time API (Application Programming Interface) that allows content to communicate with an LMS in a standard fashion. This SCORM API is described in the SCORM 1.2 Run-time Environment document.

SCORM Run-time Extension API Object

The SCORM API is typically found in a frameset of the run-time player. Many common algorithms exist for locating the SCORM API. The Aspen SCORM Run-time Extension API is a JavaScript object that is in the same window as the SCORM API and is named `API_Extended`.

Typically a course will use the SCORM API `GetAPI()` method which will return the API from this method. A better solution would be to return the window object that holds the API so you can get a handle to both API and `API_Extended`. See the example on the next page for an illustration of this approach.

Figure 4 Modified GetAPI() code to find API and API_Extended

```
var nFindAPIAttempts = 0;
var API               = null;
var API_Extended      = null;
var maxAttempts       = 500;

// change this to return a handle to the window object, not the API
function ScanForAPI(win)
{
    while ((win.API == null) && (win.parent != null) && (win.parent != win))
    {
        nFindAPIAttempts++;

        if (nFindAPIAttempts > maxAttempts)
        {
            alert("Error in finding API instance -- too deeply nested.");
            return null;
        }

        win = win.parent;
    }

    return win;
}

function GetAPI()
{
    var apiWin = null;

    if ((win.parent != null) && (win.parent != win))
    {
        apiWin = ScanForAPI(win.parent);
    }

    if ((API == null) && (win.opener != null))
    {
        apiWin = ScanForAPI(win.opener);
    }

    if ( apiWin != null )
    {
        API = apiWin.API;
        API_Extended = apiWin.API_Extended;
    }
}
```

Aspen Run-time Extensions API

The Aspen SCORM Run-time Extensions consist of two methods and ten commands. Commands are passed to the methods to perform a specific action.

Figure 5 API Methods for Aspen SCORM Extensions

| Method | Description |
|---------------------|---|
| GetNavCommandStatus | <p>This method returns the availability status of a given command.</p> <p>syntax: GetNavCommandStatus (command)</p> <p>parameter: command is a vocabulary string for a specific action; see figure 6 on the next page for a list of valid commands</p> <p>returns: boolean true/false</p> <p>example:</p> <pre>var can_nav_next = API_Extended.GetNavCommandStatus("next");</pre> |
| SetNavCommand | <p>This method performs the specified command if available.</p> <p>syntax: SetNavCommand (command)</p> <p>parameter: command is a vocabulary string for a specific action</p> <p>returns: void</p> <p>example:</p> <pre>if (can_nav_next == true) { API_Extended.SetNavCommand("next"); // navigate to next SCO }</pre> |

SCORM Run-time Extensions API Commands

These are the valid commands that can be passed as parameters to the Aspen Extended API. If an invalid command is passed to either method it is ignored.

Figure 6 Commands for Aspen SCORM Extensions API

| Command | Description |
|-------------|---|
| previous | Moves to the previous SCO in the course hierarchy unless on first SCO. |
| next | Moves to the next SCO in the course hierarchy unless on last SCO. |
| home | Moves to the first SCO in the course hierarchy. |
| study-aids | Opens the Study Aids panel if there is a license for Aspen Personalized Delivery Module. |
| progress | Opens the Progress panel. |
| search | Opens the Search panel if there is a license for Aspen Personalized Delivery Module. |
| collaborate | Opens the Collaboration panel if Collaboration Centers are licensed and the use of Collaboration Centers has been enabled for the course. |
| exit | Posts in-state data for the SCO to the LMS and closes the run-time player. |
| open-toc | Opens the Table of Contents pane if closed. Does nothing for a single-SCO course. |
| close-toc | Closes the Table of Contents pane if open. Does nothing for a single-SCO course. |

Calling `SetNavCommand` with `previous`, `next`, `home`, or `exit` will force the SCO to be unloaded. Therefore you should save and commit any SCORM data before calling `SetNavCommand` with `previous`, `next`, `home`, or `exit`. It is recommended that you also call `LMSFinish` before making this call unless you call `LMSFinish` automatically when the SCO is unloaded. Calling an Aspen extended API method does not violate the SCORM protocol, because the object you are calling with `SetNavCommand` is not the SCORM API adapter, and communications with the Aspen Extended API are outside the scope of the SCORM API specification.

Modifying the Aspen Content Player

You can modify the `imsmanifest` file to control aspects of the Aspen content player. In the example shown on the next page, the navigation bar is hidden. This example shows how to modify the tags for SCORM content created in ToolBook; for information about SCORM courses developed with a different authoring program contact technical support. To follow this example, open the `imsmanifest` file and add the lines shown with the yellow highlight under the manifest element.

```

<manifest identifier="ToolBookGeneratedSCORMContent" version="1.2"
xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:c2lcp="http://www.click2learn.com/xsd/c2l_cp_rootv1p1"
xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2 imscp_rootv1p1p2.xsd
http://www.imsglobal.org/xsd/imsmd_rootv1p2p1 imsmd_rootv1p2p1.xsd
http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd
http://www.click2learn.com/xsd/c2l_cp_rootv1p1c2l_cp_rootv1p1.xsd">

```

To hide the navigation bar, set the value of the following line equal to no:

```
<c2lcp:ShowNavBar> = no
```

In order for this change to take effect, add all of the following elements after resource element:

```

<c2lcp:ItemDataExtra>
<c2lcp:ItemData Type="Course">
<c2lcp:ItemSpecificData>
<c2lcp:CourseData>
<c2lcp:PackageProperties>
<c2lcp:CourseDisplay>
  <c2lcp:ColorNav>
    <c2lcp:Light />
    <c2lcp:Medium />
    <c2lcp:Dark />
  </c2lcp:ColorNav>
  <c2lcp:ColorData>
    <c2lcp:Light />
    <c2lcp:Medium />
    <c2lcp:Dark />
  </c2lcp:ColorData -->

<c2lcp:ShowNavBar>no</c2lcp:ShowNavBar>

<c2lcp:TOC>
  <c2lcp:MaxLevels>3</c2lcp:MaxLevels>
  <c2lcp:MaxWidth>1</c2lcp:MaxWidth>
  <c2lcp:ScrollBar>yes</c2lcp:ScrollBar>
  <c2lcp:TreeOpenIcon width="9" height="9" />
  <c2lcp:TreeCloseIcon width="9" height="9" />
  <c2lcp:TopImage width="-1" height="-1" />
  <c2lcp:BottomImage width="-1" height="-1" />
  <c2lcp:BackgroundColor>ffffff</c2lcp:BackgroundColor>
</c2lcp:TOC>
</c2lcp:CourseDisplay>
<c2lcp:Launch>
  <c2lcp:Width>797</c2lcp:Width>
  <c2lcp:Height>585</c2lcp:Height>

```

```
<c2lcp:AllowResize>no</c2lcp:AllowResize>  
</c2lcp:Launch>  
<c2lcp:GlobalData />  
</c2lcp:PackageProperties>  
</c2lcp:CourseData>  
</c2lcp:ItemSpecificData>  
</c2lcp:ItemData>  
</c2lcp:ItemDataExtra>
```