



This Guide is intended for people who want to understand the technical requirements for adding content to AgLearn. As such, it includes technical terms that may be unfamiliar to some readers.

Courseware Technical Reference Guide

Version 1.1



Contents

Introduction	4
Purpose.....	4
Audience	4
Environmental Considerations	4
Server Platform	5
Network Bandwidth	5
Network Security	5
Cross-Domain Scripting Issues.....	5
Host Platform Specifications	6
End-User Platform Specifications	6
Accessibility and Section 508 Compliance.....	10
Software Standards.....	10
Web Standards	12
Section 508 Standards Impact Map.....	14
Content Packages.....	18
Course/Content Updates.....	19
SCORM.....	19
Overview	19
The SCORM Communication API.....	22
SCORM 1.2 API Calls Supported By AgLearn+.....	23
Making The SCORM API Visible For Debugging.....	27
Cross-Domain Communication Restriction	34
Reverse Proxy.....	37
Proxlet Solution.....	37
Document Domain Manipulation	38
Troubleshooting SCORM 1.2 Content	39
AICC.....	42
Overview of AICC.....	42
AICC Required Data Format	42
The Most Common Commands:	44

The AICC [Core] group	46
Lesson_Status	47
Lesson_location	49
AICC [Objectives_Status].....	50
J_Status Values	51
Using AICC [Core_Vendor].....	53
AICC Max Normal	53
AICC Import Files.....	54
Common Issues With AICC Content.....	55
AICC Import Files Issues	57
AgLearn+ Integration Guide for Content Developers.....	59
Appendix A: List of Acronyms	65
Appendix B: Network Impact Assessment (NIA) Guidelines.....	66
Introduction	66
Network Impact Ratings.....	66
Conducting a Network Impact Assessment	67
Traffic Profiles	67
Network Friendly Application.....	68
Network Neutral Application.....	69
Network Intensive Application	70
Remediation	71
Points of Contact.....	71

Introduction

Purpose

The intent of this document is to provide developers of content for the United States Department of Agriculture (USDA) a better understanding of the USDA Information Technology (IT) environment, as well as the standards and requirements that apply to content developed for the USDA. This document is meant to be an overview and is not a substitute to any appropriate documentation regarding the specific technologies referenced herein. The technical specifications contained within this document are based on industry standards and best practices and are provided as guidance and references for designers and developers in the creation of content for the USDA. Content developers should also note that the IT environment at USDA is constantly evolving, as are industry standard technologies and best practices. Contact the USDA's AgLearn Office with questions regarding the compatibility of content or practices with the USDA IT infrastructure.

Audience

It is assumed that content developers have knowledge of and/or experience with:

- Multiple delivery environments including UNIX web servers and CD-ROMs
- Industry standard web technologies including HTML, DHTML, XML, CSS, and JavaScript
- Information architecture and file management best practices including directory structures and the use of file naming conventions
- Techniques for creating accessible content meeting the requirements for Section 508 Compliancy
- Requirements for creating Shareable Content Object Reference Model (SCORM) version 1.2 conformant content

Environmental Considerations

It is important to consider how content will be delivered, managed, and potentially reused or repurposed. All content designed and developed for distribution within the USDA must allow for a variety of environmental variances without negatively impacting the user's viewing experience. These variances may include whether or not the content is to be delivered via the USDA Intranet, the Internet, or physical digital media such as CD-ROM or DVD-ROM. Some of the factors to consider when assessing the delivery environment include:

Server Platform

All AgLearn+ servers used in staging and production are UNIX based servers and content should be developed with the UNIX platform in mind. UNIX is a case-sensitive platform, therefore case conventions and standards must be carefully followed, especially when naming and referencing files and directories.

Network Bandwidth

When delivering courses across a network, identify the expected bandwidth and how it will affect any multimedia elements contained in the course. As a general best practice, it is recommended that all content be developed assuming delivery across a low bandwidth 56kbps connection. This will allow for the same content to be delivered both to users with higher bandwidth connections, and to those without network connections via a CDROM-based delivery.

Network Security

For purposes of security when dealing with SSI content, this access has been designed to incorporate a content caching capability and code has been implemented to clear cached content from the local computer cache at the end of each session. Presently, this solution works for SSI content in the HTML, SWF, PDF, and MS Office formats. For non-SSI courseware, caching is not a concern from a security standpoint; however the performance of this content is still affected by the caching Implementation.

Cross-Domain Scripting Issues

It is a known issue within the SCORM community that courses typically cannot be delivered by an LMS if the course content resides within another domain; a domain being defined as a web URL, not an IP address. For example, “www.abc.com” is not the same as “www.xyz.com”. The source of this restriction is the JavaScript used to implement and support SCORM functionality, which is often not permitted for security reasons. It is necessary to fully understand this issue if the courseware is intended to be delivered or distributed across multiple domains. A complete description of this issue and possible solutions can be found in the document “Cross-Domain Scripting Issue, Version 1.0” from the ADL. This is not currently an issue for the USDA IT infrastructure as both the content and application reside within the same web domain, but it should be noted, as the platform is susceptible to change.

NOTE: For additional information on cross-domain scripting and SCORM, refer to the ADL website (<http://www.adlnet.org>). Additionally, cross-domain scripting is currently not allowed by USDA.

Host Platform Specifications

+ *Production Environment*

Currently, all live and approved content reside within the USDA's production environment. No courseware testing can be administered on this server. A separate staging environment is configured to support content testing for both developers and for the USDA course owners. In addition, telecommunications impact assessments are also conducted within staging. No content will be migrated to the production environment until all phases of testing have been completed on staging.

+ *Staging Environment*

Enables content developers to log into the system from external locations for the purposes of testing using generic login accounts. Team AgLearn should be contacted at TeamAgLearn@usda.gov to obtain a generic login account. These accounts are shared and rotated between vendors as required. So do not consider your account assignment a permanent assignment. In addition, the production database has been migrated back to the staging environment. This allows USDA users and administrators access to the staging environment using their regular AgLearn+ eAuthentication logins.

End-User Platform Specifications

USDA End-Users will be using any combination hardware, software, and OS to access AgLearn+ content. The most current requirements can be found after logging into AgLearn at:

<https://aglearn.usda.gov/plateau/softwarechecker/checkSoftware.do?showDetail=Y>
(login required). Below is a screen shot of the current results of using this tool on a compliant end-user machine.

Software Check Results

PLEASE READ: If you are currently using a Government furnished computer, please do not click the links below. Contact your local IT support with the information so they can assist you with any updates.

The information below contains the results from the software check:

- Microsoft Internet Explorer 7.0 -  Supported browser is detected.
 - Supported browsers:
 - Internet Explorer (Supported version is between 6.0 and 8.0.)
 - Mozilla Firefox (Supported version is 3.0.x.x and above.)
 - Safari on Mac (Supported version is 3.2 and above.)
- Java 1.6.0.24 -  You have the correct version installed and enabled.
 - Supported version is between 1.5 and 1.6.x.x.
- Flash 10.1.53.64 -  You have the correct version installed and enabled.
 - Supported version is 10.0 and above.
- Adobe Reader 8.1.0 -  You have the correct version installed and enabled.
 - Supported version is 8.0 and above.

Desktop System

Courseware Expectations

The usability of courseware content is of utmost importance, including clarity of the navigational structure, the means of navigating the content (the actual controls for navigation), accessibility both as mandated by Section 508 and as would be generally accepted in industry best practice, and any other aspects of the courseware that affects the user experience.

While the intent of the AgLearn+ is to deliver web-based training, lack of connectivity in some locations or the inaccessibility of a training room to a screener requires that all content be deliverable as both WBT and CBT. Some considerations of the dual purposing of content include limited scripting language support, file size optimization, and file type limitations. Server-side scripting languages such as .php and .asp should be avoided in favor of JavaScript, which would allow the same content to be delivered on both a CD-ROM or over a web environment without any significant changes to the code. Content must not require Internet access to any remote files (such as those residing on a vendor's host server) since external Internet access may not be available in some locations. Content should load quickly over a range of bandwidths and network speeds

including dial-up connections and LANs.

Best practices for achieving this include the optimization of file types and sizes using the following techniques:

- + *Web-native content*
For standard web-native content, which may include HTML, graphical elements, JavaScript scripting, etc, the total load not including any cached items should not exceed 80KB in size.
- + *GIFs, JPGs, and PNGs for graphical purposes*
Any images used as part of course content should be in the .gif, .jpg, or .png format. Ideally the .png format is reserved only for use when the alpha transparency feature of a .png is absolutely necessary, as some older web browsers are not capable of displaying this file format. When using any of these file types, it is recommended that their file size does not exceed 60KB in size.
- + *SWFs for content/animation/interactive graphics*
A benefit to using Flash for courseware content includes the ability to break down an animation or interaction into multiple files that load individually to a user's computer. This speeds the initial appearance of the content and allows a developer to manipulate the efficiency of the animation's loading by dictating which externally loaded .swf files load and when. When using the .swf file format for graphical animation purposes, it is recommended that each individual file be a maximum of 70KB to 100KB in size.
- + *SWFs for audio/video*
Multimedia content including audio and video should be optimized for delivery over low bandwidth connections. Currently, Macromedia Flash is the preferred format for multimedia type content being delivered over the USDA IT infrastructure. A benefit to using Flash is the ability to break down content into multiple files that load individually to a user's computer. It is recommended that no single .swf used for audio/video multimedia file be larger than 40MB in size.
- + *Other graphical/audio/video formats*
Bandwidth intensive formats including .avi, .mpeg, or other audio/video formats that may significantly negatively impact the performance of the USDA network are generally not recommended for use in courseware content. If a new technique or format outside of those recommended by the USDA AgLearn+ is desired, a vendor should deliver content to act as a "preliminary technical test" for compatibility testing with the USDA infrastructure prior to the start of development work.

+ *Bookmarking*

Book marks should be placed in the course that allow the user to either re-enter the course where they exited or to re-enter the course at a sound re-starting point such as a topic or chapter.

Accessibility and Section 508 Compliance

The following sections identify 1) the standards and 2) a functional interpretation for ensuring Section 508 compliance.

Software Standards

Standard: §1194.21(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

Interpretation: All elements that can be accessed by mouse must be keyboard accessible for non-mouse users.

Standard: §1194.21(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.

Interpretation: Use of the application must not interrupt user's accessibility functions and must adopt operating system appearance attributes

Standard: §1194.21(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.

Interpretation: Screen focus should follow a logical reading order (left to right, top to bottom)

Standard: §1194.21(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.

Interpretation: Images, animations, graphs, and audio files must have equivalent descriptions for screen reader users. Decorative images that do not convey information should not be announced by a screen Reader. Images must be used consistently throughout the application for users with cognitive disabilities.

Standard: §1194.21(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall

be consistent throughout an application's performance.

Interpretation: All buttons and labels and functions should be the same consistently throughout the course.

Standard: §1194.21(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.

Standard: §1194.21(g) Applications shall not override user selected contrast and color selections and other individual display attributes.

Interpretation: If color and contrast settings are adjustable they shall not override the user's selected contrast, color selections and other individual display attributes.

Standard: §1194.21(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.

Interpretation: The animation must not be the only way to convey information.

Standard: §1194.21(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

Interpretation: Information provided by color must also be available in text.

Standard: §1194.21(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.

Interpretation: If color and contrast settings are adjustable, a variety of settings must be available

Standard: §1194.21(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.

Standard: §1194.21(l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

Interpretation: Form fields must be explicitly labeled for assistive technology to read aloud the correct information so a user can complete the forms. Speech recognition software also relies on form field labels.

Web Standards

Standard: §1194.22(a) A text equivalent for every non-text element shall be provided (e.g., via “alt”, “longdesc”, or in the element content).

Interpretation: Images, animations, graphs, and audio files must have equivalent descriptions for users who use assistive technology. Decorative images that do not convey information should not be announced by the assistive technology. Images must be used consistently throughout the application for users with cognitive disabilities.

Standard: §1194.22(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

Interpretation: Must have accurate transcript/synchronized captions.

Standard: §1194.22(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

Interpretation: provided by color must also be available in text.

Standard: §1194.22(d) Documents shall be organized so they are readable without requiring an associated style sheet.

Interpretation: A page must be readable and functional without the developer’s style sheet.

Standard: §1194.22(e) Redundant text links shall be provided for each active region of a server-side image map.

Interpretation: a method to create links for each active region of a server-side image map shall be provided

Standard: §1194.22(f) Client-side image maps shall be provided instead of server-side image maps except where the regions be defined with an available geometric shape.

Interpretation: application shall be able to provide client-side image maps that include the creation of alt text for both images and hot spots.

Standard: §1194.22(g) Row and column headers shall be identified for data tables.

Interpretation: Data table column and row headers must be identified.

Standard: §1194.22(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

Interpretation: Tables containing more than one level of information should utilize markup to specify what category and subcategory the data belongs to.

Standard: §1194.22(i) Frames shall be titled with text that facilitates frame identification and navigation.

Interpretation: Frame Titles must be descriptive so the content of the frame is obvious to aid screen reader navigation.

Standard: §1194.22(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Standard: §1194.22(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

Interpretation: In the event that a web page or element within a web page cannot be made to comply with the above conditions, an alternative compliant page should be created. This page should be kept up to date and always reflect the content of non-complaint page.

Standard: §1194.22(l) When web pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

Interpretation: If scripting language is utilized to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

Standard: §1194.22(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with 1194.21 (a) through (l).

Interpretation: If the content of a web page requires an applet or plug-in, links must be provided to download plug-ins required to view file types that will not display in the browser.

Standard: §1194.22(n) When electronic forms are designed to be completed on line, the form shall allow people using assistive technology to access the information, field elements and functionality required for completion and submission of the form, including all directions and cues.

Interpretation: Form fields must be explicitly labeled for screen readers to read aloud the correct information so a user can complete the forms. Speech recognition software also relies on form field labels.

Standard: §1194.22(o) A method shall be provided that permits users to skip repetitive navigation links.

Interpretation: Any page with repetitive links (such as navigation bars or links in a header or footer) must also contain a method to skip past these links.

Standard: §1194.22(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Interpretation: The user must be alerted that time is about to expire and be allowed to request more time.

Section 508 Standards Impact Map

Software Standards	Blind	Low Vision	Deaf/ Hard of Hearing	Mobility	*Other
1194.21 (a) Keyboard access	x			x	
1194.21 (b)					
1. Accessibility feature – Narrator	x				
2. Accessibility feature - High contrast		x			
3. Accessibility feature- Sound Sentry			x		
4. Accessibility feature- Sticky Keys				x	
1194.21 (c) On screen focus	x	x		x	

1194.21 (d) Sufficient information for user interface elements	x			x	
1194.21 (e) Bitmap image consistency					x
1194.21 (f) Textual information through operating system functions	x				
1194.21 (g) User color and contrast settings		x			
1194.21 (h) Animation alternative	x	x			
1194.21 (i) Color dependency	x	x			x
1194.21 (j) Color selection variety		x			x
1194.21 (k) Flickering elements					x
1194.21 (l) Label form fields	x				

Web Standards	Blind	Low Vision	Deaf/ Hard of Hearing	Mobility	*Other
1194.22 (a) Text equivalents	x				
1194.22 (b)					
1. Synchronized captions			x		
2. Audio description	x	x			
1194.22 (c) Color dependence	x	x			x
1194.22 (d) Style sheet dependence	x	x			x
1194.22 (e) Text links for server side image maps	x			x	
1194.22 (f) Client side image maps instead of server side	x			x	
1194.22 (g) Simple data tables: Identify row and column headers with <scope>	x				

1194.22 (h) Complex data tables: Associate data cells to row and column headers	x				
1194.22 (i) Descriptive titles for frames.	x				
1194.22 (j) Flickering elements					x
1194.22 k. Equivalent text only page, only when necessary.	x	x	x	x	
1194.22 (l) Functional text for scripts	x				
1194.22 (m) Provide plugin links when required	x				
1194.22 (n) Label form fields	x				
1194.22 (o) Skip repetitive navigation links	x	x		x	
1194.22 (p) Timed response	x	x		x	x

Content Packages

When submitting content for upload into AgLearn, please submit an AgLearn Course Content Submission form on the AgLearn Central SharePoint site (at [https://portal.usda.gov/sites/AGLearn/Lists/DRAFT AgLearn Course Content Submission/](https://portal.usda.gov/sites/AGLearn/Lists/DRAFT%20AgLearn%20Course%20Content%20Submission/)) Note: a login is required to access this site.

AgLearn Central > BETA AgLearn Course Content Submission > New Item

BETA AgLearn Course Content Submission: New Item

OK Cancel

Attach File Spelling... * indicates a required field

Submission Date *	9/22/2011
Sponsoring Agency/Office: *	AMS Sponsoring agency or office
Course Title *	<input type="text"/> The name of the course being submitted.
AgLearn Item ID *	<input type="text"/> The Item ID to be used in AgLearn. It should start with the acronym for the agency or office.
Has a conformance memo been added? *	<input type="checkbox"/> An AgLearn content conformance memo must be included. Click on the paper clip in the upper left-hand corner to add files. Added files are listed at the bottom of the form.
Course Description: *	<input type="text"/> Click for help about adding basic HTML formatting.
Mandatory course?:	No
Expected Duration (in hours): *	1.0
Item type: *	<input checked="" type="radio"/> Web-based <input type="radio"/> Specify your own value: <input type="text"/>
Subject Area(s): *	<input type="text"/> For the AgLearn catalog listings. Examples include Human Resources, Accounting, and IT Security.

Once complete, the progress of the submission package can then be tracked on the Course Integration Tracking page.

Agency/Office Name	Course Title	Submitted Date	Est. ETA to Prod	Date posted to Prod	Status	Being
APHIS	ADR - Civil Rights Training 2011	6/22/2011	6/29/2011	6/29/2011	Complete	Eva C
ARS	REE Individual Development Plan (IDP) Training	6/17/2011	6/24/2011	7/22/2011	Completed	Eva C
ARS	Ree Individual Development Plan (IDP) Training (paper version)	7/18/2011	7/25/2011	7/22/2011	Completed	Eva C
ASCR	No FEAR Act Refresher Training	7/22/2011	7/29/2011	7/25/2011	Completed	Eva C
DA/DM	Sustainable Acquisition Training	7/14/2011	7/21/2011	7/28/2011	Complete	Eva C
DA/DM	USDA Annual SCI Refresher Briefing	8/17/2011	8/24/2011	8/23/2011	Completed	Eva C
FS	6140 FS National (Session 1) Role of the FS Supervisor and Manager in HRM / Civil Rights	9/15/2011	9/22/2011		On staging server for final review	Eva C
FS	6140 FS National (Session 2) Performance Management	9/15/2011	9/22/2011		There is a problem; see Comments	Eva C
FSA	Introduction to Hiring Reform for FFAS Supervisors and Managers	8/17/2011	8/24/2011		On staging server for final review	Eva C
FSA	HP Quality Center Defect Management Training	8/17/2011	8/24/2011		On staging server for final review	Eva C
NASS	NASS Project Management Manual Overview	9/12/2011	9/19/2011		On staging server for final review	Eva C
NRCS	NRCS Cultural Transformation Awareness Training	9/1/2011	9/8/2011	9/2/2011	In Production	Eva C
OCFO	FMMI Business Objects Viewer - User Training	8/22/2011	8/29/2011	9/8/2011	Complete	Eva C
OCIO	Veteran Employment Training for Federal Hiring Managers	7/5/2011	7/12/2011	8/30/2011	Completed	Eva C
OCIO	Veteran Employment Training for Federal Human Resource Professionals	7/5/2011	7/12/2011	8/30/2011	Completed	Eva C
OCIO	Principles of Plain Language Training	7/11/2011	7/18/2011		There is a problem; see Comments	Eva C

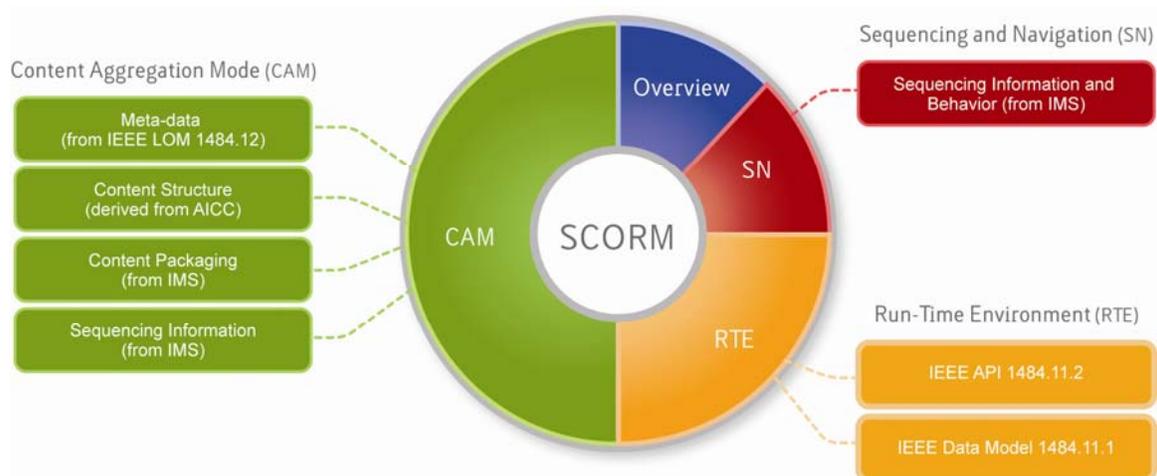
Guidance for the “Fast Track” content integration process can be found in the Shared Documents/Content Integration folder in AgLearn Central.

SCORM

Overview

The **Sharable Content Object Reference Model (SCORM)** is a model that references a set of interrelated technical specifications and guidelines designed to meet the Department of Defense's high-level requirements for Web-based learning content. In a nutshell, SCORM was developed by the Advanced Distributed Learning (ADL) initiative for the Department of Defense as a means of constructing and presenting "learning objects".

It is important to note that SCORM is basically an amalgamation of several different pre-existing standards. The SCORM standard is comprised of the following standards governing the setup and presentation of online content:



As shown above, SCORM treats each individually referenced specification as a separate "book". With the release of version 1.2, the SCORM was divided into three books (Overview, CAM, and RTE). SCORM 2004 added the Sequencing and Navigation specification. Each of these books are available from the [ADL website](#).

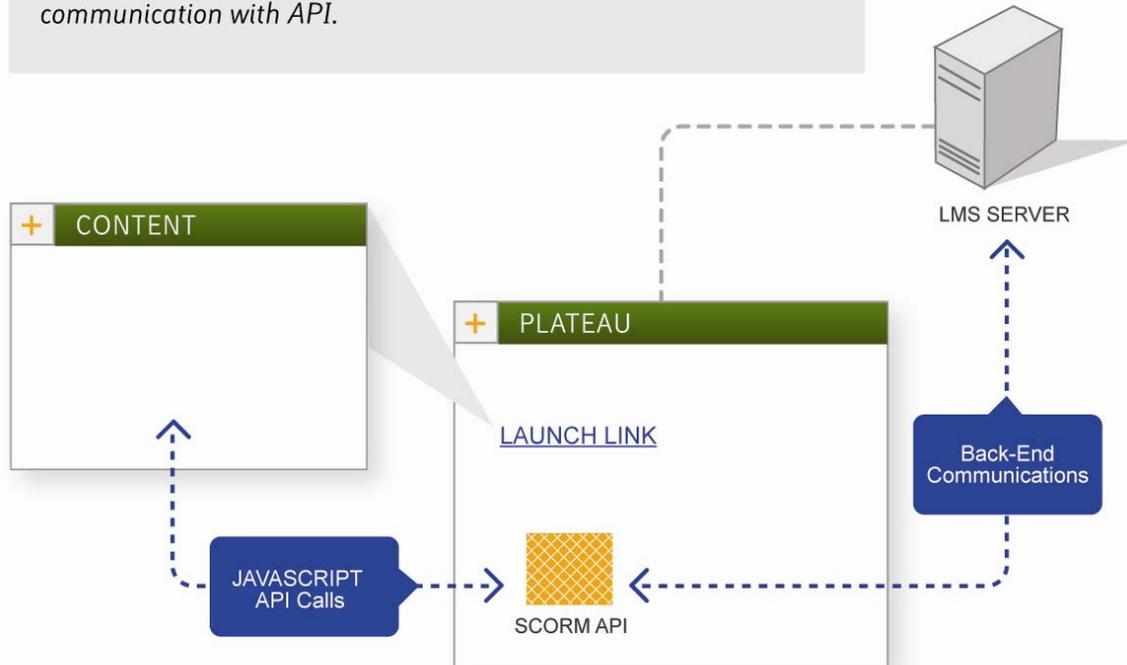
- + **Overview**
Covers areas of the CAM, RTE, and SN books at a high level.
- + **Content Aggregation Model - CAM**
The SCORM Content Aggregation Model contains guidance for identifying and aggregating resources into structured learning content. This standard describes the content of the `imsmanifest.xml` file which details the course structure, supporting meta-data and sequencing information (SCORM 2004 only). The `imsmanifest.xml` is the file used by an LMS to import SCORM content.
- + **Run-Time Environment - RTE**
The SCORM Run-Time Environment includes guidance for launching, communicating with and tracking content in a web-based environment. This book was originally derived from the run-time environment functionality defined in AICC's CMI001 Guidelines for Interoperability. It details the responsibilities of both the content and the LMS regarding the communication of tracking data.
- + **Sequencing & Navigation (SCORM 2004 only)**
The SCORM SN book describes how SCORM conformant content may be sequenced through a set of learner-initiated or system-initiated navigation events. This defines how a learner may proceed through a particular course.

NOTES:

- *This site will discuss the SCORM 1.2 and SCORM 2004 CAM, RTE, and SN as they pertain to AgLearn+.*
- *The Run-Time Environment (content communication) standard for SCORM 1.2 was taken from AICC Appendix B (API method) so the SCORM 1.2 and AICC data models are essentially the same.*
- *SCORM 2004 diverged slightly from the original AICC spec, but much of the functionality remains the same.*

The SCORM Communication API

Content searches self then opens window for the SCORM API. When the API is found, the content initializes the API then sets/gets values to/from API. When finished, the content closes communication with API.



Initially, the SCORM communication model was borrowed from the AICC Appendix B (API) model and thus uses many of the same data elements discussed in the AICC portion of this site. As shown above, SCORM communication is processed by a client-side API provided by the LMS. The content object (SCO - sharable content object) communicates with the API through a series of SCORM-defined javascript function calls. The API processes the functions and handles the communication with the LMS. SCORM 2004 introduced some additional functionality which moved beyond the original AICC communication model. For a comparison of SCORM 1.2 and SCORM 2004, please see the [SCORM 1.2 vs SCORM 2004](#) page of this site.

The basic idea is any SCORM compliant LMS will provide an API that understands a set of SCORM-defined javascript functions. All the messy details of communicating with the LMS are handled by the API, thus allowing SCORM content to communicate with any SCORM compliant LMS simply by using the predefined API functions.

NOTES:

- *If SCORM content is NOT hosted under the same DNS (someserver.yourcompany.com) as the LMS, the browser will prevent the SCO from communicating with the API. Please see the **Cross-Domain Restriction** page of this site for more information.*
- *For a listing of the functions and parameters supported by AgLearn+'s API, please see the appropriate SCORM technical section of this document.*

SCORM 1.2 API Calls Supported By AgLearn+

As discussed in the [API Communication Overview](#) page, SCORM content communicates with an LMS via a client-side API. This API is designed to process a set of functions (defined by SCORM), and relay the information to the LMS. It is important to note the communication portion of SCORM 1.2 was borrowed from the AICC standard and thus shares roughly the same data model as AICC. For a list of the data elements used in AICC communication, please refer to the [Standard AICC DATA Values](#) page of this site.

LMSInitialize()	The content must call this function before calling any other API function. It indicates to the LMS system that the content is going to communicate. The LMS can take any initialization steps required in this function. For instance, this can contact the CMI via AICC messaging to pull down the cmi.core variables for quick access.
LMSGetValue()	<p>This function is used to pass data from the CMI to the content. Only one value is returned for each call. The category and/or element is named in the parameter.</p> <p>The following LMSGetValue parameters are supported by AgLearn+:</p> <ul style="list-style-type: none">• cmi.core._children - returns "student_id, student_name, lesson_location, credit,

lesson_status, entry, score, total_time, exit, session_time".

- **cmi.core.student_id**
- **cmi.core.student_name**
- **cmi.core.lesson_location** - 255 char string typically used for bookmarking
- **cmi.core.credit** - returns "no-credit" or "credit"
- **cmi.core.lesson_status** - indicates SCO completion - returns one of the following: "passed", "completed", "failed", "incomplete", "browsed", or "not attempted"
- **cmi.core.entry** - returns "ab-initio", "resume", or ""
- **cmi.core.score_children** - returns "raw, min, max"
- **cmi.core.score.raw** - the raw score value
- **cmi.core.score.max** - the maximum possible score - if not set, assumed 100
- **cmi.core.score.min** - the minimum possible score - if not set, assumed 0
- **cmi.core.total_time** - returns the sum of all a learner's session times
- **cmi.core.lesson_mode** - returns "browse", "normal", or "review"
- **cmi.suspend_data** - 4096 char string usually used for progress data
- **cmi.launch_data** - 4096 char string used to pass data to the SCO on launch from the LMS
- **cmi.objectives_children** - returns "id, score, status"
- **cmi.objectives_count** - returns the number of objectives currently stored for this SCO.
- **cmi.objectives.n.id** - returns the objective identifier where "n" is the objective number
- **cmi.objectives.n.score_children** - returns "raw, min, max "
- **cmi.objectives.n.score.raw** - the raw objective score value
- **cmi.objectives.n.score.min** - The minimum possible score - if not set, assumed 0
- **cmi.objectives.n.score.max** - the maximum possible objective score - if not set, assumed 100
- **cmi.objectives.n.status** - indicates if the learner has completed the objective - returns one of the following: "passed", "completed", "failed",

	<p>"incomplete", "browsed", or "not attempted"</p> <p>The following are legitimate element names, but are write-only. These would therefore return "" and set an error: 404 - element is write-only.</p> <ul style="list-style-type: none">• cmi.core.exit• cmi.core.session_time
LMSSetValue()	<p>This function is used to pass data from the content to the LMS. The parameter indicates which category or element is being set. Only one value may be set with a single function call.</p> <p>The following LMSSetValue parameters are supported by AgLearn+:</p> <ul style="list-style-type: none">• cmi.core.lesson_location - 255 char string usually used for bookmarking• cmi.core.lesson_status - indicates SCO completion - accepts "passed", "completed", "failed", "incomplete", "browsed", "not attempted"• cmi.core.score.raw - the raw score value• cmi.core.score.max - the maximum possible score - if not set, assumed 100• cmi.core.score.min - the minimum possible score - if not set, assumed 0• cmi.core.exit - indicates how the learner left the SCO - accepts "time-out", "suspend", or "logout"• cmi.core_session_time - Time spent in the SCO - HHHH:MM:SS.SS• cmi.suspend_data - 4096 char string usually used for progress data• cmi.objectives.n.id - the objective identifier where "n" is the objective number• cmi.objectives.n.score.raw - the raw objective score value• cmi.objectives.n.score.min - The minimum possible score - if not set, assumed 0• cmi.objectives.n.score.max - the maximum possible objective score - if not set, assumed 100

- **cmi.objectives.n.status** - indicates if the learner has completed the objective - accepts "passed", "completed", "failed", "incomplete", "browsed", or "not attempted"

The following are legitimate element names, but are read-only. These would therefore return "false" and set an error: 403 - element is read-only:

- **cmi.core.student_id**
- **cmi.core.student_name**
- **cmi.core.credit**
- **cmi.core.entry**
- **cmi.core.total_time**
- **cmi.core.lesson_mode**
- **cmi.launch_data**

The following elements are legitimate element names, but are keywords for multiple elements. These would therefore return "" and set the error code 402 - "Invalid set value, element is a keyword":

- **cmi.core._children**
- **cmi.core.score._children**
- **cmi.objectives._children**
- **cmi.objectives._count**
- **cmi.objectives.n.score._children**

LMSCommit()

Any values set using the LMSSetValue command are stored for delivery. This function submits any values not yet sent to the LMS.

LMSFinish()	The content must call this function before it terminates, if it successfully called LMSInitialize at any point. It signals to the LMS that the content has finished communicating. The content may not call any API function except LMSGetLastError after it calls LMSFinish
LMSGetLastError()	This function provides content with a way of assessing whether or not any given API call was successful, and if it was not successful, what went wrong. This routine returns an error code from the previous API call. Each time an API function is called (with the exception of this one), the error code is reset in the API. The content may call this any number of times to retrieve the error code, and the code will not change until the next API call.
LMSGetErrorString()	This function returns a textual description of the error represented by an error code number.
LMSGetDiagnostic()	This function would return an LMS-specific error description based on an error code number.

NOTE: For more detailed information, please refer to SCORM Run Time Environment document included in the Resources section of this guide.

Making The SCORM API Visible For Debugging

The SCORM API can be made visible in AgLearn+, allowing a user to monitor the communication between the content and the API.

To make the applet visible, edit the *main_online_frameset.jsp* page in AgLearn+.

Open the *main_online_frameset.jsp* page in a text editor such as notepad and remove the two slashes (("//") before the following line:

```
//debugMode = "1";
```

NOTES:

- It may be necessary to clear the application server cache after making changes to JSP files
- If you are running AgLearn+ on a Weblogic application server, the `main_online_frameset.jsp` will be in the `elms.war` package which is inside the `elms.ear` file. In addition, you will need to remove the servlet-mapping for the `main_online_frameset.jsp` in the `web.xml` file (also in the `elms.war` package).
- *If you are running AgLearn+ on a Weblogic application server, the `main_online_frameset.jsp` will be in the `elms.war` package which is inside the `elms.ear` file. In addition, you will need to remove the servlet-mapping for the `main_online_frameset.jsp` in the `web.xml` file (also in the `elms.war` package).*

SCORM Import

SCORM content usually contains files which are used by an LMS to import the course definitions. The primary import file is the `imsmanifest.xml` which may or may not have supporting xml files. It is important to note that the actual content is not imported into the LMS, just the definition or setup of the content (the Content Package and Content Object definitions).

A SCORM `imsmanifest.xml` file may be divided into three primary parts.

- The Resources Section
- The Organizations Section
- Supporting Meta-data

The Resources section is composed of a single `<resources>` element which contains one or `<resource>` elements. Individual `<resource>` elements define individual pieces of content which may be either an Asset or a SCO. An asset is simply a piece of content which does not contain code to communicate SCORM data. A SCO is content which contains code for SCORM communication.

Resource elements contain the following attributes:

<code><resource></code> Attribute	Description
	Identifies the resource as either an "asset" or a

adlcp:scormtype	"sco"
identifier	Unique ID for the resource. Used in the Organizations section to reference the resource.
type	Defines the type of resource. Always defined as "webcontent".
href	Required for SCO resources. This is the filepath to the resource.

```
<resources>
  <resource identifier="R_S100004" type="webcontent"
    adlcp:scormtype="sco" href="Course01/Lesson01/sco01.htm">
  </resource>
  <resource identifier="R_D1" adlcp:scormtype="asset"
    type="webcontent" xml:base="Course01">
    <file href="SCOFunctions.js"/>
    <file href="APIWrapper.js"/>
  </resource>
</resources>
```

The Organizations section is composed of a single <organizations> element which typically contains a single, but may contain multiple, <organization> elements. An organization element defines a course structure using one or more <item> elements. An item may map to a <resource> resulting in a content object, or an item may contain other item elements resulting in a folder in the course structure. In AgLearn+, an item referencing a "sco" type resource is imported as a SCORM type content object and an item referencing an "asset" type resource is imported as a content object that is configured to "mark complete on launch".

Item elements contain the following attributes:

<item> Attribute	Description
Identifierref	Ties the item to a resource through the resource identifier attribute.

Identifier	Unique ID for the item.
parameters	Used to define URL parameters to be passed to the content.
isvisible	Indicates if the item is visible in the LMS course structure.

```
<organizations default="B0">
  <organization identifier="B0">
    <title>Maritime Navigation</title>
    <item identifier="B110" isvisible="true">
      <title>Steering &amp; Sailing Rules</title>

      <item identifier="S110001" identifierref="R_S110001">
        <title>Conduct of Vessels in any Condition of Visibility</title>
      </item>

      <item identifier="S110002" identifierref="R_S110002">
        <title>Conduct of Vessels in Sight of One Another</title>
      </item>
    </item>
  </organization>
</organizations>
```

The above sample would result in the following structure:

- Steering & Sailing Rules
 - [Conduct of Vessels in any Condition of Visibility](#)
 - [Conduct of Vessels in Sight of One Another](#)

Supporting Meta-data is defined in a <metadata> element and may be applied to individual <resource>, <organization>, or <item> elements. Meta-data elements may be defined in the imsmanifest.xml file itself or in supporting xml documents identified in the <metadata> element. In the following example, metadata for SCO R_S110002 is defined in a supporting xml document named sco03.xml.

```
<resource identifier="R_S110002" type="webcontent"
  adlcp:scormtype="sco" href="Course01/Lesson01/sco03.htm">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
    <adlcp:location>Course01/Lesson01/sco03.xml</adlcp:location>
  </metadata>
  <file href="Course01/Lesson01/sco03.htm" />
</resource>
```

A metadata element can contain a wide array of information most of which is irrelevant to an LMS. AgLearn+ only pulls the following elements from supporting meta-data.

- Title - Defined in a <title> element under the <general> element
- Description information - defined in a <description> element under the <general> element.
- Educational Objective - The objective is pulled from the <description> element under a <classification> element but only if the <purpose> element for the <classification> contains a <value> element defined as "Educational Objective".

```
<general>
  <title>
    <langstring>Conduct of Vessels in any Condition of Visibility</lan
  </title>
  <description>
    <langstring>
      Discusses general rules of operation for vessels on inland
      waters. Topics discussed include: Look-out, Safe Speed, Coll
      Channels, Traffic Separation.
    </langstring>
  </description>
</general>

<classification>
  <purpose>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Educational Objective</langstring
    </value>
  </purpose>
  <description>
    <langstring>
      This Sharable Content Object will give the student a basic
      understanding of the conduct of vessels in any condition of
      visibility.
    </langstring>
  </description>
</classification>
```

AgLearn+ Content Package				
AgLearn+ Field	Manifest Document	Section	Data Element	Condition

Title	imsmanifest.xml	Organizations	Organization title	
Description	Organization metadata	General	Description	
Objective	Organization metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose"
Folder label	imsmanifest.xml	Organization	Item.title	If Item data element contains sub Items
Content object label	imsmanifest.xml	Organization	Item.title	If Item identifier ref attribute maps to an asset or SCO resource.
AgLearn+ Content Objects (SCO-level)				
AgLearn+ Field	Manifest Document	Section	Data Element	Condition

ID	imsmanifest.xml	Resources	Resource. identifier	auto-generates content object ID from the first 15 characters of the manifest identifier, an underscore and then the first 12 characters of the resource identifier. <first 15 of manifest identifier>_<first 12 of resource identifier>
Title	resource metadata	General	Title	
Description	resource metadata	General	Description	
Launch path	imsmanifest.xml	Resources	href	If Resource is referenced in the <organization>
Objective	resource metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose". The metadata for Purpose>Source>

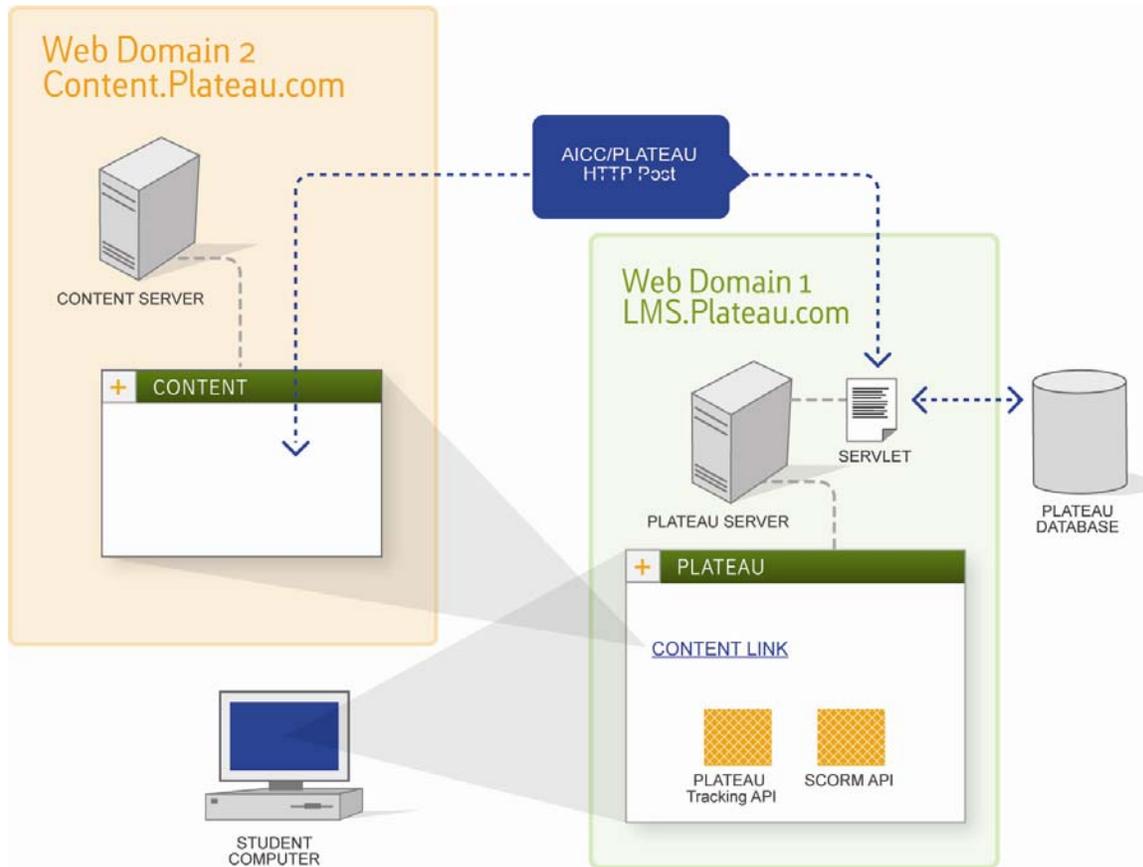
				langstring should be "LOMv1.0".
--	--	--	--	---------------------------------

The AgLearn+ SCORM Import Assistant prompts the user for a zip file, and will not process any remote file over 1MB. If you have a SCORM course zip file that is over 1MB, and would like to use the import process from a local file, simply extract all the xml files into a new Zip file and use the new file with the import assistant. If you wish to import a zip from a URL, the zip file will need to be hosted on a web server that shares the same DNS as the LMS server (something.somethingelse.com), and a fully-qualified path (http://...) to the zip file must be provided.

SCORM 2004 introduced sequencing elements to the imsmanifest file. Please see the [SCORM Sequencing page](#) of this site for more details.

Cross-Domain Communication Restriction

Content can communicate with AgLearn+ using one of two technologies; a direct HTTP Post or via a client-side API. AICC content uses direct HTTP posts to send information to AgLearn+. SCORM content uses a client-side API. The Plateau Tracking Functions can use either direct HTTP posts or a client-side API. When content is using a client-side API for communication, that content must be hosted under the same DNS as the API it is attempting to use. If the content and the API are hosted under separate DNS names, the browser will prevent them from communicating. It is important to note that this is not a AgLearn+ specific issue. It is a restriction that is inherent in the SCORM communication standard.



In the diagram above, the content and the API's are hosted under separate DNS names, preventing them from communicating with one another. Under this scenario, only direct HTTP posts may be used for communication.

If it is necessary to host your content under a separate DNS, several solutions exist to work around the cross-domain restriction. ADL, the writers of the SCORM spec, have compiled a collection of potential solutions for the cross-domain issue. This paper is available below and can also be found on the ADL web site (<http://adlnet.org>):

[ADL Cross-Domain Scripting Issue v2.0](#)

[ADL Cross-Domain Proxy Solution](#)

Solutions discussed in the ADL paper fall under three categories:

- Combination LMS and content solution

- Network architecture solution
- LMS-based solution

Reverse Proxy

One common network architecture solution to work around the cross-domain restriction is the reverse proxy. This is essentially a proxy server that sits in front of the LMS and any content servers in your environment. All requests for the LMS or content are routed through the reverse proxy and are forwarded to the appropriate servers based on the context of the request.

For example, if your network contained the following servers:

- LMS = `http://lms.customer.com/elms/learner/login.jsp`
- Content Server 1 = `http://content.customer.com/content1/course_x/sco_x.htm`
- Content Server 2 = `http://content2.customer.com/content2/course_y/index.htm`
- A reverse proxy server (`http://revProxy.customer.com`) could be added to your network with the following rules:
 - route any `"/elms"` request to `"http://lms.customer.com/elms"`
 - route any `"/content1"` request to `"http://content.customer.com/content1"`
 - route any `"/content2"` request to `"http://content2.customer.com/content2"`
 - This would allow a user to use the following reverse proxy requests to access the LMS or content:
 - LMS = `http://revProxy.customer.com/elms/learner/login.jsp`
 - Content Server 1 = `http://revProxy.customer.com/content1/course_x/sco_x.htm`
 - Content Server 2 = `http://revProxy.customer.com/content2/course_y/index.htm`

The browser would see the content and the LMS as coming from the same DNS and thus the cross-domain restriction would not apply.

The [ADL Cross-Domain Proxy Solution](#) paper discusses the implementation and configuration of the reverse proxy solution in detail.

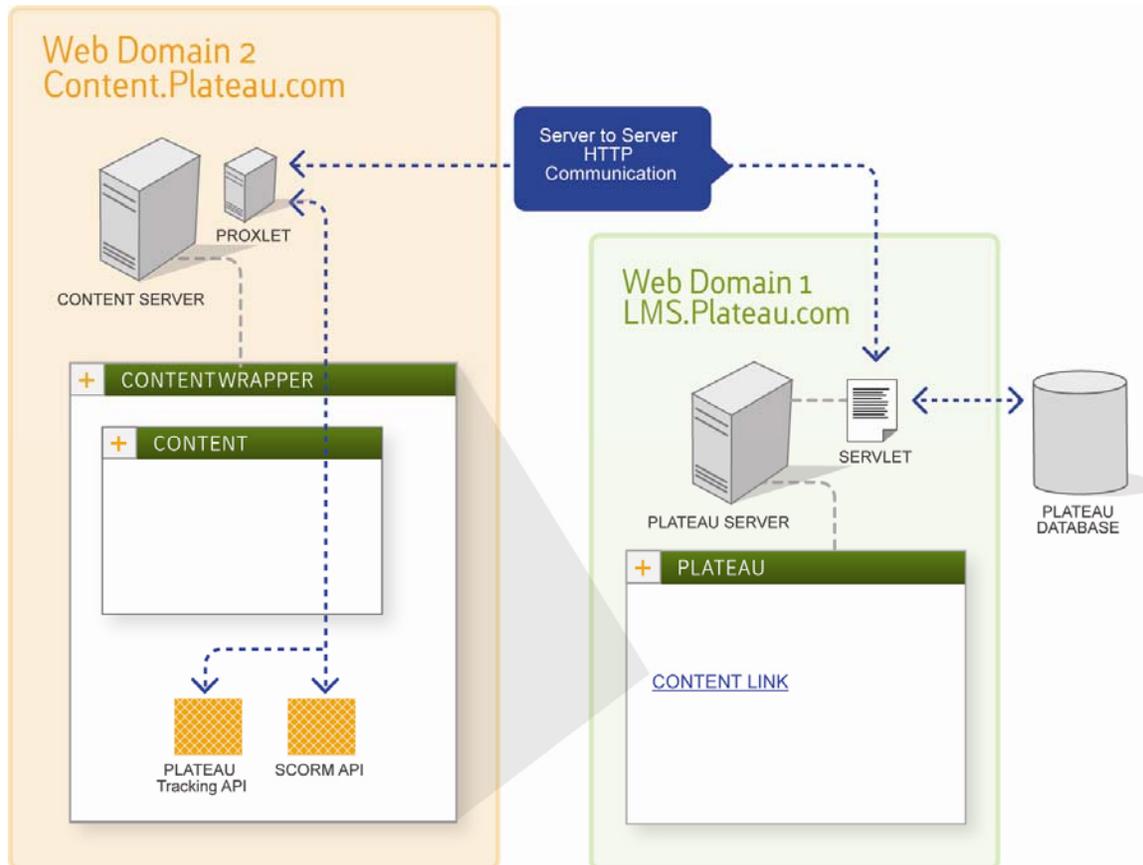
Proxlet Solution

In AgLearn+ 5.5 an entirely LMS-based solution (no changes required in the content) was introduced. This solution resolves the cross-domain restriction by placing the communication APIs on the content server and relies on a communication proxy (proxlet) on the content server to relay tracking data back to AgLearn+ (section 1.3.8 of the ADL document). It can be thought of as placing part of the LMS, needed by the content for communication, on the content server. This solution is enabled via the LMS configuration files, and requires additional settings at the content object or question object level. This solution resolves the cross-domain restriction for the following APIs:

- Plateau Tracking

- SCORM
- PEXAM ([external questions for internal exams](#))

[Click Here](#) for information on deploying this solution.



The diagram above shows the LMS-based solution introduced in AgLearn+ 5.5. The content and the APIs are loaded into a wrapper. The APIs communicate back to AgLearn+ via a communication proxy running on the content server.

Document Domain Manipulation

AgLearn+ 4.1 introduced a cross-domain configuration setting in the LMS configuration framework. This setting allows a user to set the domain of the SCORM and Plateau Tracking API's to a root domain. For instance, the diagram above has the API's hosted under LMS.PLATEAU.COM, so AgLearn+ could be configured to host the API's under PLATEAU.COM (it is not possible to specify just COM as a domain).

Once AgLearn+'s API's are configured to a root domain, it is possible to use javascript to manipulate the domain of the content. The following code can be used to hard code the domain of the content:

```
<script language="javascript">  
document.domain="Plateau.com";  
</script>
```

Or this code can be used to dynamically set the domain to the top root-level domain. For instance, if the domain was MY.CONTENT.SERVER.PLATEAU.COM, this code would set the domain to PLATEAU.COM.

```
<script language="javascript">  
midDotLoc = document.domain.lastIndexOf('.');  
dotLoc = document.domain.lastIndexOf('.',midDotLoc-1);  
document.domain=document.domain.substring(dotLoc+1);  
</script>
```

Once the AgLearn+ API domain and the content domain have been set to a common root, the cross-domain restriction no longer applies. and the content will be able to access the APIs.

Troubleshooting SCORM 1.2 Content

The following are some issues that have frequently arisen and suggested solutions.

Content is unable to find the SCORM API

When SCORM content is launched, the first thing it does is to search for the LMS communication piece, an object called "API". The content searches all the frames in it's own window first, and if it does not find the API object, it attempts to search in it's opener window. In AgLearn+, the API is hosted in the learner application, so the content will have to search the opener window to find the API. There are several reasons why the content may not be able to find the API.

- Make sure the content is specified as a "SCORM" type content object in AgLearn+ (use "Browser" type for AgLearn+ v.4.2 and earlier). The SCORM 1.2 API is not available to AICC, Browser, SCORM2004 or Document type content objects.
- If the content is hosted under a separate DNS name (server name) from the LMS, the content will not be able to access the API. The browser, for security

reasons, will prevent the content from searching in the opener window and will usually give an "Access Denied" error. For more information, please see the [cross-domain restriction](#) page of this site.

- The code that looks for the API may need to be altered. Typically there will be an APIWrapper.js file that contains functions provided by ADL (writers of the SCORM spec). The functions that locate the API are "getAPI" and "findAPI". The getAPI function calls the findAPI function and specifies a window to search. As mentioned above, the content will first search the content window, and then the opener window (AgLearn+ student piece). The getAPI call to search the opener window is usually: `theAPI = findAPI(window.opener);`
- If the SCORM content is using frames, this function call will not make sense to the browser because it will look for the opener of a particular frame. It may be necessary to fix this line of code to search the opener of the frameset as follows: `theAPI = findAPI(top.window.opener);`
- Likewise, if the content is using an opener window to format the content's browser window, the content will be two levels out from the LMS and thus the proper call would need to be: `theAPI = findAPI(window.opener.opener);`

Not sure if the content is communicating

There are two ways to confirm content communication. The first method is to check the learner record in the AgLearn+ Administrator.

- Log-in to AgLearn+ Administrator.
- Open the Learners>Online Status tab.
- Click the Object Details link to examine the details for the content object. Values updated by SCORM include Total Time, Score, Finished, Complete, Objective Finished, and Objective Complete.

This method should allow you to see if the content is communicating, but it will not show you all the SCORM data being stored in the AgLearn+ Database. Since AICC and SCORM share a common data model, it is possible to construct a simple AICC request to obtain all the content data stored in the database. The following page can be used to retrieve AICC or SCORM data from AgLearn+. If no data is recorded for the content, please read the "Content is unable to find the SCORM API" section above.

Content is communicating but is not marked complete

Content completion is tied to the SCORM Lesson_Status value. AgLearn+ only grants credit for a Lesson_Status value of "passed", however, many content vendors only send a value of "completed". It is possible to configure AgLearn+ to grant credit for a sco sending a Lesson_Status value of "completed". Please see the Lesson_Status section

on the [AICC Common Issues](#) page for details. If the Lesson_Status and score values are not being updated, check the [Max Normal](#) setting for the item.

Objective data is not recorded

SCORM content has the ability to send Objective data to AgLearn+; however, the objectives must exist in AgLearn+, and must be associated with the content object. If the objectives are listed in the SCORM import files, they will be created and assigned during the import process. If the objectives are listed, but are not being generated during import, please see the objectives section on the [SCORM import page](#) and confirm the import file is properly formatted.

SCORM import is generating errors

Please see the [SCORM Import page](#).

AICC

Overview of AICC

The **Aviation Industry CBT (Computer-Based Training) Committee (AICC)** is an international association of technology-based training professionals. The AICC develops guidelines for the aviation industry in the development, delivery, and evaluation of CBT and related training technologies. <http://www.aicc.org>

Purpose: AICC provides a standard for communication between training content and learning management systems.

Explanation: As an LMS, AgLearn+ is focused on two facets of the AICC guidelines:

- + *Communication between content and an LMS (AgLearn+).*
The Communication guidelines set well defined rules for passing information from content to an LMS and vice-versa. The guidelines specify the method by which information should be passed, and the format in which the information should be communicated (analogous to saying we will communicate by telephone and in French). The AICC outlined two means of communicating with online content. These are defined in Appendix A and Appendix B of the [AICC CMI Guidelines for Interoperability](#) document.
- + *Content setup within an LMS (AgLearn+'s AICC Import Assistant).*
The AICC guidelines define a set of files used to provide an LMS with course information. These files may be used by an LMS to import the course. It is important to note that the actual content is not imported or exported, just the definition or setup information for the content; the item (component) and content object definitions.

AICC Required Data Format

AICC communication is performed by HTTP posts comprised of a series of name/value pairs (see the table below).

When AICC content is launched from an LMS, two parameters are appended to the launch url: AICC_SID and AICC_URL. The content retrieves these parameters from the launch path and uses them to communicate back to the LMS (AgLearn+). The AICC_URL is the URL to which AICC data is posted. The AICC_SID is a unique identifier for the learner, course, and content object. In an AICC post, the AICC_SID is returned to the LMS under the "session_id" parameter.

In AgLearn+, the AICC launch parameters are composed of the following data:

- AICC_URL = <LMS protocol>://<LMS DNS>:<LMS port>/<LMS application context>/PwsAicc - for example if a learner logged into the AgLearn+ user application using https://lms.company.com:7008/plateau/user/login.jsp then the AICC_URL would be https://lms.company.com:7008/plateau/PwsAicc.
- AICC_SID = "C"<studCpntID>"M"<modID>"S" where the studCpntID is a unique numeric identifier for the student and item, and the modID is a unique numeric identifier for the content object. for example, if the studCpntID = 175 and the modID = 21, then the AICC_SID would be C175M21S.

NOTE: Each student will have a unique AICC_SID for any given content object

AICC parameter	Description
Command	Any valid AICC HTTP command (i.e., PutParam or GetParam)
Version	AICC Spec Version (AgLearn+ is certified under version 2.2)
session_id	Unique Session Identifier passed on the launch URL (AICC_SID value from the launch url)
AU_password (This parameter is optional)	Assignable Unit (content object) specific password
AICC_Data	

Data specific to the command. If the content was sending information to the LMS, this is the parameter under which the data would be sent. The AICC_Data value must be URL encoded.

AICC HTTP Post Parameters

Usage Rules:

- The AICC_Data value is URL-encoded.
- The Name/value pairs can appear in any order.
- If an optional value is to be omitted, the name must also be omitted.
- The name of each parameter is not case sensitive.

The Most Common Commands:

"PutParam" and "GetParam" (PutParam sends data to AgLearn+, and GetParam retrieves data from AgLearn+). Others can be found in the [CMI Guidelines for Interoperability](#) document under Appendix A.

Examples:

Click on the examples below to see the LMS response data for GetParam and putParam commands.

- http://courseware.plateausystems.com/plateau/PwsAicc?Command=GetParam&version=2.2&session_id=C175M21S
- [http://courseware.plateausystems.com/plateau/PwsAicc?Command=PutParam&Version=2.2&Session_id=C175M21S&aicc_data=\[core\]%0D%0Alesson_location%3Dend%0D%0Alesson_status%3DP%0D%0Ascore%3D87%0D%0Atime%3D00:23:15](http://courseware.plateausystems.com/plateau/PwsAicc?Command=PutParam&Version=2.2&Session_id=C175M21S&aicc_data=[core]%0D%0Alesson_location%3Dend%0D%0Alesson_status%3DP%0D%0Ascore%3D87%0D%0Atime%3D00:23:15)

Tips

- %0D is the hex value for carriage return
- %0A is the hex value for line feed
- %3D is the hex value for =
- %26 the hex value for &

Standard AICC_Data Values

An AICC post is comprised of a series of name/value pairs (see the [Required Data Format](#) page). The AICC_Data parameter is the name/value pair that contains the

specific data being sent from the content to the LMS. The AICC_Data parameter is also comprised of name/value pairs which define the information being sent (see the table below). The AICC_Data value is url-encoded.

Group	Parameters	Description
[Core] (required)	Lesson_status	Used for passing module status to AgLearn+
	Lesson_location	Used to store bookmarking data
	Score	Score (one per content object)
	Time	HHHH:MM:SS.SS (accumulative value- adds to previous time in AgLearn+)
[Objectives_Status]	J_ID.1	Objective Identifier (must match Objective ID in AgLearn+)
	J_Score.1	Score for the objective
	J_Status.1	Objective Status
[Core_Lesson]	Data is undefined and may be unique to each lesson	Open-ended element used to store content specific data (see the Using Core lesson page).

NOTE: A carriage return and line feed must be included after each name/value pair and group tag (i.e. [core]).

Usage Rules

- The AICC_Data value is URL-encoded.
- The Name/value pairs can appear in any order.
- If an optional value is to be omitted, the name must also be omitted.
- The name of each parameter is not case sensitive.
- The [core] group and all of its parameters are required.

Example:

```
[core]
lesson_location=end
lesson_status=pass
score=87
time=00:23:15
[core_lesson]
This is sample text for the core_lesson parameter
```

URL-encoded, the string would look like this:

[\[core\]%0D%0Alesson_location%3Dend%0D%0Alesson_status%3Dpass%0D%0Ascore%3D87%0D%0Atime%3D00:23:15%0D%0A\[core_lesson\]%0D%0AThis%20is%20sample%20text%20for%20the%20core_lesson%20parameter](#)

The full AICC post would look like this:

[http://courseware.plateausystems.com/plateau/PwsAicc?Command=PutParam&Version=2.2&Session_id=C175M21S&aicc_data=\[core\]%0D%0Alesson_location%3Dend%0D%0Alesson_status%3Dpass%0D%0Ascore%3D87%0D%0Atime%3D00:23:15%0D%0A\[core_lesson\]%0D%0AThis%20is%20sample%20text%20for%20the%20core_lesson%20parameter](http://courseware.plateausystems.com/plateau/PwsAicc?Command=PutParam&Version=2.2&Session_id=C175M21S&aicc_data=[core]%0D%0Alesson_location%3Dend%0D%0Alesson_status%3Dpass%0D%0Ascore%3D87%0D%0Atime%3D00:23:15%0D%0A[core_lesson]%0D%0AThis%20is%20sample%20text%20for%20the%20core_lesson%20parameter)

Tips

- %0D is the hex value for carriage return
- %0A is the hex value for line feed
- %3D is the hex value for =
- %26 is the hex value for &
- %20 is the hex value for a space

The AICC [Core] group

The Core group contains the main data elements for your content. The Core group and all of its parameters are required when sending an AICC post.

Group	Parameters	Description
[Core] (required)	Lesson_status	Used for passing completion status to AgLearn+
	Lesson_location	Used to store bookmarking data
	Score	Score (one per content object)
	Time	HHHH:MM:SS.SS (accumulative value- adds to previous time in AgLearn+)

Lesson_Status

Lesson_Status is used to determine the completion status of the content module.

Acceptable values for Lesson_Status are as follows:

- + *Passed (or p)*
 Necessary number of objectives in the lesson were mastered, or the necessary score was achieved. Student is considered to have completed the lesson and passed.
- + *Completed (or c)*
 The lesson may or may not be passed, but all the elements in the lesson were experienced by the student. The student is considered to have completed the lesson.
- + *Failed (or f)*
 The lesson was not passed. All the lesson elements may or may not have been completed by the student.
- + *Incomplete (or i)*
 The lesson was begun but not finished.
- + *Not attempted (or n or na)*

This is the initial lesson_status value set by the LMS. Not attempted means that the student did not even begin the lesson.

AgLearn+ interprets the values as follows:

AICC Lesson_Status	AgLearn+ Finished <i>(Viewed but not mastered all the material)</i>	AgLearn+ Complete <i>(Earned Credit)</i>
Passed	yes	yes
Completed	yes	no
Failed	yes	no
Incomplete	no	no
Not attempted	no	no

Each Lesson_Status value may be followed by one of the following flags:

- + *Time_out (or T)*
 This indicates the lesson ended because the lesson has determined an excessive amount of time has elapsed.
- + *Suspend (or S)*
 This indicates the student leaves the lesson with the intent of returning to it later at the point where he/she left off. The LMS will send a resume lesson_status flag the next time the content is launched.
- + *Logout (or L)*
 This indicates that the student logged out of the LMS from within the lesson instead of returning to the LMS to log out. The intent is to require re-authentication from the student.

Rules:

- Only the first letter of the lesson_status value is processed, so "P", "pass", and "passed" would all be treated the same.
- The Lesson_Status and Lesson_Status Flag are separated by a comma (for example: P,L).
- The flag information is not passed back as part of the Getparam response. The only flags possible when passing information back to the content are the ab initio and resume flags. For more information, please refer to the [CMI Guidelines for Interoperability](#) document.

Lesson_location

Bookmarking information can be stored using the AICC Lesson_Location value. Each new Lesson_location post overwrites the existing value. The Lesson_Location field is limited to 255 characters.

Score

Indication of the performance of the student during his/her last session in the content object. This score may be determined and calculated in any manner that makes sense to the program designer. For instance, it could reflect the percentage of objectives complete, it could be the raw score on a multiple choice test, or it could indicate the number of correct first responses to the embedded questions in the content object.

The score can be a stand-alone value, or may be followed by two values: a maximum and minimum. Where the maximum is the largest score the student could have achieved with the interactions experienced and the minimum is the smallest score that the student could have achieved with the interactions experienced.

Format:

Decimal number or blank for all three values. If three values are passed, they are separated by commas, and the order is significant: Score, Maximum, Minimum

Score Examples:

SCORE= 79

Probably a percentage result.

Score = 8.0,10.0,0.0

Raw score of 8 with a maximum of 10.

AgLearn+ would show a percentage score of 80%.

Time

Accumulated time of all the student sessions in the content.

format:

HHHH:MM:SS.SS Integer number representing hours, followed by a colon, an integer from 00 to 59 representing minutes, followed by a colon and a decimal or integer from 00 to 59.99 representing seconds. Three numbers, separated by colons, are always required, even if only seconds or minutes are represented.

Examples:

TIME=00:29:00

Student spent 29 minutes in lesson.

time=01:27:00

Student spent 1 hour 27 minutes in lesson.

AICC [Objectives_Status]

The Objectives_Status group allows the content to store information related to the lesson's learning objectives. To link an AICC objective with an AgLearn+ objective, create an objective in AgLearn+ and assign the AICC objective ID as the AgLearn+ "objective" field (not the auto-generated objective ID value). The newly created AgLearn+ objectives can then be assigned to the AICC content module that contains the objectives. AgLearn+ will then update the objective status using the J_Status and J_Score values passed for that objective.

Group	Parameters	Description
[Objectives_Status]	J_ID.1	The lesson-specific objective identifier.
	J_Score.1	Score for the objective
	J_Status.1	Objective Status

NOTE: The Objectives Status group can contain multiple objectives differentiated by a unique numeric parameter extension (i.e. J_ID.3 or J_ID.9).

J_Score Values

The objective score parameter indicates the score obtained by the student after each attempt to master an objective. A maximum and minimum value may accompany the score. The maximum values indicate the largest score the student could have achieved and the minimum value is the smallest achievable score.

- A semicolon separates multiple attempts with the latest attempt listed first.
- Commas separate maximum and minimum values.

J_Score Examples:

J_Score.2=87

The student scored an 87 (assumed to be 87%).

J_Score.2=2,3

The student scored 2 out of a maximum possible score of 3.

J_Score.4=9.5,10,0;6.5,10,0

On the student's first attempt, they scored a 6.5 out of a maximum of 10 and a minimum of 0, and on their second attempt, they scored a 9.5 out of a maximum of 10 and a minimum of 0.

J_Status Values

The objective status parameter indicates the completion status of the objective. There are five possible statuses:

- + *Pass (or p or pass)*
The student has mastered the objective.
- + *Complete (or c)*
The student has gone through all segments of the lesson related to the objective. She may or may not have passed.
- + *Fail (or f)*
Failed may be followed by a comma and an integer number indicating the number of times the objective has been failed.
- + *Incomplete (or i)*
The student has not gone through all the segments of the lesson related to this objective.

- + *Not attempted (or n or NA)*
The student has not gone through any of the segments of the lesson related to this objective.

- + *Browsed (or b)*
The student launched the lesson with a CMI mode of Browse on the initial attempt.

Objectives_Status Usage Rules:

- The Objectives_Status values are URL-encoded and passed as part of the AICC_Data parameter.
- Objectives_Status data is processed as part of either a "PutParam" or a "PutObjectives" command.
- The Name/value pairs can appear in any order.
- If an optional value is to be omitted, the name must also be omitted.
- The name of each parameter is not case sensitive.
-

Objectives_Status Examples:

```
[Objectives_Status]
J_ID.1=Obj1
J_Status.1=Passed
J_Score.1=87
J_ID.2=Obj2
J_Status.2=Failed
J_Score.2=33
```

URL-encoded, the string would look like this:

```
[Objectives_Status]%0D%0AJ_ID.1%3DObj1%0D%0AJ_Status.1%3DPassed%0D%0A
J_Score.1%3D87%0D%0AJ_ID.2%3DObj2%0D%0AJ_Status.2%3DFailed%0D%0AJ_S
core.2%3D33
```

Tips:

- %0D is the hex value for carriage return
- %0A is the hex value for line feed
- %3D is the hex value for =

Using AICC [Core_Vendor]

The Core_Vendor group is used to pass data to the lesson on launch. Core_Vendor data can be entered in the content object's "AICC Support" tab within AgLearn+.

Core_Vendor data is only passed from the LMS to the content, and cannot be reset by the content. This data may be set automatically by the AICC import assistant from the "Core_Vendor" parameter of the AU import file. This data group is limited to 4096 characters.

[Content Objects](#)

[| Simple Search](#) | [Advanced Search](#) | [Add New](#) | [Help](#) |

[> Search](#) > [Search Results](#) > [Edit AICC Support](#)

Content Object ID: toolkit aicc testpage
Title: aicc_testpage

[Summary](#) | [Launch Method](#) | [Objectives](#) | **[AICC Support](#)** | [Cross Domain](#)

Edit the AICC Support for the Content Object

AICC Support:

[Apply Changes](#) [Reset](#)

Using AICC [Core_Lesson]

The Core_Lesson group is contained in the AICC_Data section of a PutParam command. Core_Lesson contains data that is created by the content and stored by the LMS to be passed back to the content the next time it is run. Typically this data is used to store learner progress data. Any new Core_Lesson data will overwrite the existing data. Core_Lesson is limited to 4096 characters.

AICC Max Normal

Value: 1 to 99, if null it defaults to 1. The Max Normal value may be set manually on the "Online Settings" tab of the item (component), or it may be set automatically by the AICC import assistant from the "max_normal" parameter in the CRS import file. The Max Normal is the maximum number of assignable units (content objects) that may be taken for credit simultaneously. That is, this value indicates how many content objects in an item (component) are allowed to be incomplete at any given time. When the number of incomplete content objects exceeds the Max Normal value, subsequent launches of AICC content will be with credit="no credit" (AICC Lesson_Status and Score

values cannot be updated).

For example, if an item has 5 content objects and the max normal value for the item is set to 3, a student can access up to 3 content objects without completion. If they access and do not complete a fourth content object, any AICC content object they access afterwards will be launched with Credit="No Credit", and the Lesson_Status and Score values will be locked.

It is important to note the completion value used for the Max Normal calculation is AgLearn+'s completion status, not AICC's. As a result, the Max Normal calculation is based on all content objects in an item whether they are AICC content objects or not (this also includes exam objects). However, only AICC (and SCORM) content objects are affected if the Max Normal value is exceeded.

AICC Import Files

In addition to developing a communication guideline, AICC also defined a method of importing content definitions into an LMS. It is important to note that the actual content is not imported, just the definition or setup of the content (the item (component) and content object definitions). Importing the same files twice will produce two sets of data, so any edits that need to be made after importing the content definitions should be made to the actual records in AgLearn+.

AgLearn+ supports the following AICC data files:

The Course file (*.crs)

This file contains course-level information about the course as a whole.

The Assignable Unit file (*.au)

This file contains technical information relating to all the assignable units (content objects) in the course (i.e. content object launch url).

The Descriptor file (*.des)

This file contains descriptive information relating to every course element in the course (i.e. content object description). It is used as the basic cross-reference file showing the correspondence of system generated IDs with user-defined IDs for every element.

The Course Structure file (*.cst)

This file contains the basic data on the content structure of the course. It includes all of the assignable units (content objects) and blocks (folders) in the course.

AICC import files may be used with the AICC import assistant in AgLearn+ to automatically generate the item and content object records for a given course. It should be noted that the AICC import assistant does not do anything that cannot be done manually through the Admin user interface.

AICC import files may be generated for existing online items in AgLearn+ by using the "AICC Export" option on the "Online Settings" tab of an item. The exported files contain the definitions of the item and all associated content objects. These export files can then be imported to another LMS, bringing in the structure of the original item.

Common Issues With AICC Content

Issue	Resolved
Lesson Status Issue	Configurable work around added in AgLearn+ 4.1.1
AICC Import Files Issues	

Lesson_Status Issue

The Lesson_Status value is passed as part of the [Core] group of the AICC_Data parameter (see the [Core] Grouppage of this site). Lesson_Status is used to determine the completion status of the content object. Acceptable values for Lesson_Status are as follows:

- + *Passed (or p)*
Necessary number of objectives in the lesson were mastered, or the necessary score was achieved. Student is considered to have completed the lesson and passed.
- + *Completed (or c)*
The lesson may or may not be passed, but all the elements in the lesson were experienced by the student. The student is considered to have completed the lesson.
- + *Failed (or f)*
The lesson was not passed. All the lesson elements may or may not have been completed by the student.
- + *Incomplete (or i)*
The lesson was begun but not finished.

- + *Not attempted (or n or na)*
 This is the initial lesson_status value set by the LMS. Not attempted means that the student did not even begin the lesson.
- +
- + *AgLearn+ interprets the values as follows:*

AICC Lesson_Status	AgLearn+ Finished (Viewed but not mastered all the material)	AgLearn+ Complete (Earned Credit)
Passed	yes	yes
Completed	yes	no
Failed	yes	no
Incomplete	no	no
not attempted	no	no

Some content providers only use Lesson_Status values of "completed" and "incomplete". This presents an issue because AgLearn+ only grants credit if a Lesson_Status value of "passed" is sent (this due to an interpretation difference of the AICC guideline definitions of "Completed" and "Passed" listed above).

Configuration for AICC Lesson_Status Work Around

In 4.1.1 and above, AgLearn+ may be configured to grant credit for a Lesson_Status value of "completed". Users can configure a keyword for one of the fields listed below, that will trigger AgLearn+ to treat "completed" as "passed".

- **Item > Source ID** - If this field is used, the trigger will be applied to all content objects in the item.

- **Content Object > Developer Tool** - Applied to the individual content object only.
- **Content Object > Build Company** - Applied to the individual content object only.
- In AgLearn+ 4.1.1, the completion triggers may be established using the following page:
- **<address of your plateau server + context path>/pwa/sysadmin/configuration_edit.jsp**
- In AgLearn+ 4.2 and above, the configuration is found under the **SysAdmin>Configuration>AICC**

Note that several triggers will already be configured. To establish a new trigger, copy and paste a complete mapping block for the appropriate field name and edit the "name" value.

For example:

```
<ContentBuildCompany name="TOOLBOOK" ignoreCase="yes">  
<LessonStatus vendor="C" aicc="P"/>  
</ContentBuildCompany>
```

This block establishes the trigger "TOOLBOOK" for the Build Company field, so if a content object's build company field contains the string "TOOLBOOK", then AgLearn+ will treat an incoming lesson_status value of "completed" as "passed" and grant credit for the object.

NOTE: The keyword is case sensitive; the ignoreCase variable applies to the lesson_Status value being communicated.

NOTE: Please make note of the following conditions:

- If using Build Company as the trigger, you can have non-trigger data in either Item Source or Developer Tool.
- If you are using Developer Tool as a trigger, you can have non-trigger data in Item Source, but if an invalid trigger string exists in Build Company, the completion is reset.
- If you are using Item Source, you cannot have invalid trigger strings in either Build Company or Developer Tool.

AICC Import Files Issues

The following issues will cause errors during the AICC import process:

- **CST file** - No empty strings should exist as member place holders. If a block has only three members, then only three members should be listed. (fixed in version 4.2)
- **AU file**- If any values are present for the "command_line" parameter in the AU file, the import process produces a content player type content object rather than an aicc type object. All "command line" parameter values should be removed from the AU file prior to import. (AICC interpretation difference)

AgLearn+ Integration Guide for Content Developers

This document contains all the information a content developer needs to successfully integrate their content with the AgLearn+ LMS. It includes information on the supported AICC and SCORM data elements and identifies integration points that may pose potential issues. [Download the guide here.](#)

AICC Supported Data Elements

[CORE]

Lesson_Status

Lesson_Location

Score (raw, min, max)

Time

Exit Flag

[CORE_LESSON]

[OBJECTIVES_STATUS]

Objective.n.status

Objective.n.score (raw,min,max)

Objective.n.id

[CORE_VENDOR]

For more information on the above data elements, please see the following pages from the content integration guide:

[core] group

<http://content.plateausystems.com/ContentIntegration/content/AICC/01-02-03.htm?standAlone=true>

[core_lesson] group

<http://content.plateausystems.com/ContentIntegration/content/AICC/01-02-06.htm?standAlone=true>

[objectives_status] group

<http://content.plateausystems.com/ContentIntegration/content/AICC/01-02-04.htm?standAlone=true>

[core_vendor] group

<http://content.plateausystems.com/ContentIntegration/content/AICC/01-02-05.htm?standAlone=true>

AICC Import Files:

AgLearn+'s AICC import assistant supports the CRS, CST, DES, and AU files.

SCORM Supported Data Elements

This information can be found on the content integration guide.

List of supported SCORM 1.2 data elements:

<http://content.plateausystems.com/ContentIntegration/content/SCORM/scormSupport.html?standAlone=true>

List of supported SCORM 2004 data

elements:http://content.plateausystems.com/ContentIntegration/content/SCORM/scormSupport_2004.html?standAlone=true

SCORM Supported CAM Data:

AgLearn+ Content Package (Course-level)				
AgLearn+ Field	Manifest Document	Section	Data Element	Condition
Title	imsmanifest.xml	Organizations	Organization title	
Description	Organization metadata	General	Description	

Objective	Organization metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose"
Folder label	imsmanifest.xml	Organization	Item.title	If Item data element contains sub Items
Content object label	imsmanifest.xml	Organization	Item.title	If Item identifierref attribute maps to an asset or SCO resource.

AgLearn+ Content Objects (SCO-level)

AgLearn+ Field	Manifest Document	Section	Data Element	Condition
ID	imsmanifest.xml	Resources	Resource.identifier	auto-generates content object ID from the first 15 characters of the manifest identifier, an underscore and then the first 12 characters of the resource identifier.
Title	resource metadata	General	Title	

Description	resource metadata	General	Description	
Launch path	imsmanifest.xml	Resources	href	If Resource is referenced in the <organization>
Launch parameters	lmsmanifest.xml	Item	parameters	Additional url parameters for a given SCO.
Objective	resource metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose". The metadata for Purpose>Source>langstring should be "LOMv1.0".

Potential Integration Issues

Applies to:	Potential Integration Issue
Content Import	
SCORM 1.2 SCORM 2004	Metadata Schema Requirement – AgLearn+ version 5.8 requires the imsmanifest to contain at least one metadata tag containing the adl

	<p>schemaversion.</p> <pre><metadata> <schema>ADL SCORM</schema> <schemaversion>1.2</schemaversion> </metadata></pre>
AICC	<p>AU Command_Line Parameter – The AU import file should not contain any command_line parameter values. AgLearn+ will not generate a valid AICC object for any assignable unit with a command_line value</p>
AICC	<p>ORT file – The AgLearn+ AICC import assistant does not support .ORT files.</p>
<p>Content Launch and Initialization</p>	
SCORM 1.2	<p>Finding the SCORM API – AgLearn+ opens content into its own window so the content must search under the opener window to locate the SCORM API.</p>
AICC SCORM 1.2	<p>Content Window size and format – AgLearn+ does not allow a user to format the content window size or appearance. AICC content is launched via a standard hyperlink. SCORM 1.2 content is launched from AgLearn+ using a window.open function call with the following settings (toolbar=1,location=0,directories=0,status=1,menubar=1,scrollbars=1,resizable=1)</p>
<p>Content Communication</p>	

<p>AICC SCORM 1.2</p>	<p>Lesson_Status (cmi.core.lesson_status) – AgLearn+ only grants credit for a lesson_status value of “P” (“passed” under SCORM). AgLearn+ offers a fix for content that only sends a Lesson_Status value of “completed”. Refer to the content integration guide for additional information http://content.plateausystems.com/ContentIntegration/content/AICC/01-02-09.htm?standAlone=true</p>
<p>AICC SCORM 1.2</p>	<p>Exit Flag (cmi.core.exit) – AgLearn+ will exit a learner from the AgLearn+ application when an exit flag value of “logout” is received.</p>
<p>AICC SCORM 1.2</p>	<p>Mastery_Score (cmi.student_data.mastery_score) - AgLearn+ versions prior to 5.8 SP2 do not support Mastery_Score. AgLearn+ support for mastery_score only includes passing the value to the content. The AgLearn+ LMS relies on the content to determine if a learner has passed or failed a module.</p>
<p>AICC SCORM 1.2</p>	<p>Time Tracking – AgLearn+ treats every time value sent in an AICC post or a SCORM commit independently, and updates the total time value of the module. If multiple time values are sent to AgLearn+ in a single content session, all will be added to the total time value for the module.</p>
<p>AICC SCORM 1.2</p>	<p>Objective Data Tracking - AICC and SCORM Objective data is not tracked unless the objectives exist in the AgLearn+ database and are associated with the content module. The objective IDs for all incoming AICC or SCORM objective data must match existing objectives in the AgLearn+ database.</p> <p>AgLearn+’s SCORM import assistant generates and associates objectives from the imsmanifest metadata, but the AICC import assistant does not generate objective records.</p>

Appendix A: List of Acronyms

- ADL** – Advanced Distributed Learning
- API** – Application Program Interface
- CBT** – Computer Based Training
- COTS** – Commercial Off the Shelf
- GUI** – Graphical User Interface
- IT** – Information Technology
- LCMS** – Learning Content Management System
- LMS** – Learning Management System
- SCO** – Shareable Content Object
- USDA** – United States Department of Agriculture
- RTE** – Run Time Environment
- SCORM** – Shareable Content Object Reference Model
- WBT** – Web Based Training

Appendix B: Network Impact Assessment (NIA) Guidelines

Introduction

The Network Impact Assessment (NIA) Guidelines is to guide the course developers and the agencies to understand and to follow in developing courses from a network perspective. The two main objectives of the NIA is to ensure that the Department's Universal Telecommunications Network (UTN) infrastructure maintains the appropriate capacity in order to support the business functions of the agencies, and to forecast the response time a user will experience during the period of the course session.

When the course contains video content and/or the course material has been developed differently from previous courses, a telecommunications network impact assessment (NIA) is required. This NIA is required during the beginning of development when a sample course has been created. The sample course should be an accurate, functional representation of the final course.

Network Impact Ratings

The following table provides the ratings given for a course's NIA.

Rating Categories	Data Exceeds the 1 Mbps threshold	Infrastructure constrained by T1 segment(s)
Network-Friendly	Data < 10%	Data < 10%
Network-Neutral	10% < Data < 50%	10% < Data < 50%
Network-Intensive	50% < Data	50% < Data

The courses are rated in three different categories: **network-friendly**, **network-neutral**, and **network-intensive**. The thresholds are rated based upon transmission of data exceeding a 1 Mbps threshold and the network demands of an infrastructure constrained by T1 (1.544 Mbps) segment(s). The T1 constraint is used to represent the limiting bandwidth availability of field office sites. Any course with a network-intensive rating will be referred to the UTN Technical Review Board (TRB) for additional consideration and approval. The purpose of the TRB is to help define, establish and operate an end-to-end enterprise telecommunications network configuration management process for significant technical changes that may impact the UTN, including steps to authorize and track those changes. By definition, courses that receive a rating of network-friendly are not require to follow this process. However, courses that receive network-neutral or

network-intensive ratings are further reviewed. If the course was rated as network-neutral or network-intensive, the following information is needed for further analysis:

- User demographics of the course
- Course frequency
- Duration and length of the course

Conducting a Network Impact Assessment

The NIA is performed from a network perspective, inferring that the amount of traffic generated includes not only the application data, but also the network protocols encapsulating the original application data stream as well as any processing overhead (i.e. encryption). The traffic generated by the course is captured using the **OPNET tool suite**. The OPNET tool suite provides the capability to capture session traffic, packet by packet and analyze the packet by packet trace in terms of application and network performance characteristics.

When the NIAs are conducted, the interest is in the network throughput between a client and a host using one second intervals. For example, the video content in course material may use either bulk download (non-streaming) or streaming technology. Since bulk downloads transfers all the data at the beginning of the course, this typically has a higher network throughput that results in a rating in the Network-Intensive category. To stream the data, the host platform would need to have the capability to stream the data. If possible, it is also recommended for course development to choose a protocol or a presentation method that limits the bandwidth below the 1 Mbps threshold. Some of the other key variables to consider when developing a course includes minimizing the use of any high volume files such as portable document formats (PDFs), MS Word, MS Excel, high resolution video, large image files and other such files that result in large instantaneous downloads.

The OPNET tool suite used by NTSO is not accessible to the agencies, but agencies can purchase the tool suite if the agency finds it beneficial for them. Also, free software such as “ethereal” can be used to capture a traffic session. The capture session file can be sent to NTSO staff for the assessment. If a course needs an assessment, the agency can coordinate with the AgLearn team to post the course on staging AgLearn in order for the NTSO staff to assess.

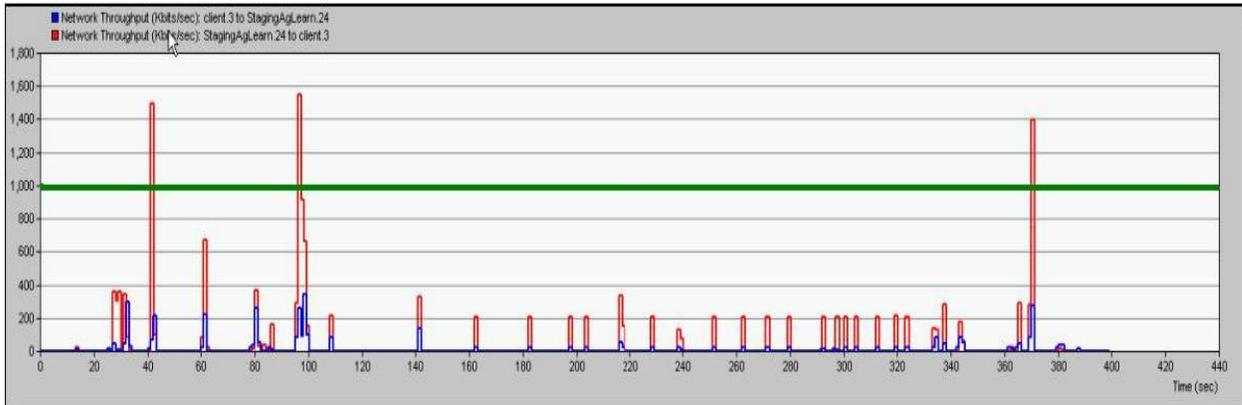
The following are three NIA scenarios for each NIA rating category to help you further understand how to maintain a network-friendly course.

Traffic Profiles

The traffic profile shown below illustrates the traffic flow generated between the Client and the Server. The blue line represents the network traffic generated from the Client to the Server, and the red line represents the network traffic generated from the Server to

the Client. The green line displays the threshold of 1 Mbps. In addition to the network throughput presented in the graph, the table shown below presents the amount of network throughput further profiled by categorizing the actual values of the traffic demands.

Network Friendly Application



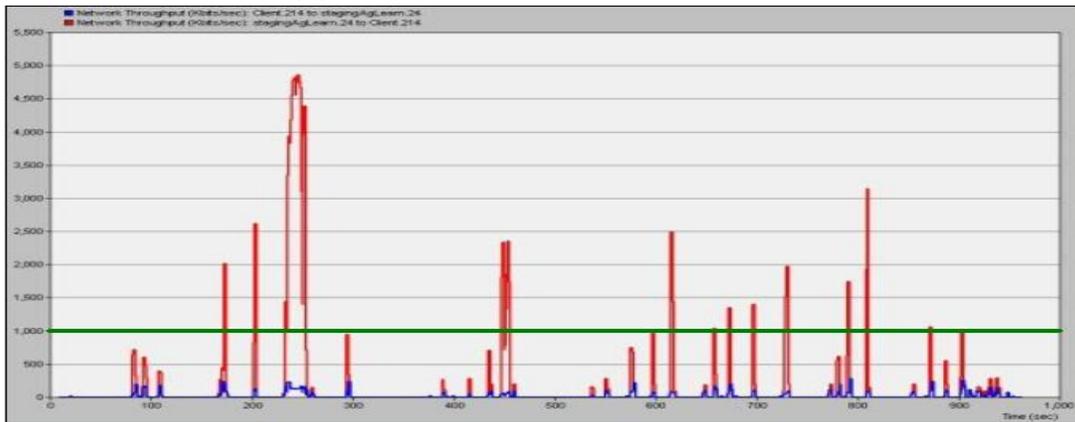
Network Throughput

	Client -to- stagingAgLearn		stagingAgLearn -to- Client	
Throughput Range	Throughput Instances	Throughput Portion (%)	Throughput Instances	Throughput Portion (%)
Total	78		65	
< 512 kbps	78	100	59	90.8
512 kbps < 1 Mbps	0	0	3	4.6
1 Mbps < 1.5 Mbps	0	0	2	3.1
1.5 Mbps < 2 Mbps	0	0	1	1.5
2 Mbps < 3 Mbps	0	0	0	0
3 Mbps < 4 Mbps	0	0	0	0
4 Mbps < 5 Mbps	0	0	0	0
> 5 Mbps	0	0	0	0

Network Throughput Demands

Notice that out of the 65 traffic demands, 3 of the traffic demands exceed the 1 Mbps threshold. This accounts for 4.6% of the traffic that exceed the threshold. By definition, this course would be rated as **Network Friendly since it occurs less than 10% of the time**. An example of a network friendly application can be one with the HTTP containing small network data exchange. For example, the graph above contains the total of 2.3 MB of network data.

Network Neutral Application



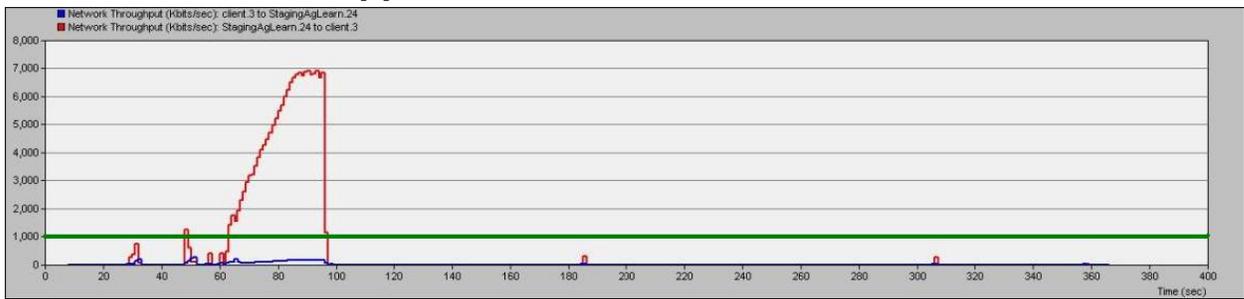
Network Throughput

Throughput Range	Client -to- stagingAgLearn		stagingAgLearn -to- Client	
	Throughput Instances	Throughput Portion (%)	Throughput Instances	Throughput Portion (%)
Total	227		204	
< 512 kbps	227	100	147	72.1
512 kbps < 1 Mbps	0	0	20	9.8
1 Mbps < 1.5 Mbps	0	0	9	4.4
1.5 Mbps < 2 Mbps	0	0	6	2.9
2 Mbps < 3 Mbps	0	0	5	2.5
3 Mbps < 4 Mbps	0	0	7	3.4
4 Mbps < 5 Mbps	0	0	10	4.9
> 5 Mbps	0	0	0	0

Network Throughput Demands

Notice that out of the 204 traffic demands, 37 of the traffic demands exceed the 1 Mbps threshold. This accounts for 18.1% of the traffic that exceed the threshold. By definition, this course would be rated as **Network Neutral since it occurs more that 10% of the time but less the 50% of the time**. An example of a network neutral application can be a course containing Adobe Flash Player with the minimum number instances that exceed the 1 Mbps threshold. The example above contains the total of 18 MB of network data.

Network Intensive Application



Network Throughput

Throughput Range	Client -to- stagingAgLearn		stagingAgLearn -to- Client	
	Throughput Instances	Throughput Portion (%)	Throughput Instances	Throughput Portion (%)
Total	67		62	
< 512 kbps	67	100	25	40.3
512 kbps < 1 Mbps	0	0	2	3.2
1 Mbps < 1.5 Mbps	0	0	3	4.8
1.5 Mbps < 2 Mbps	0	0	3	4.8
2 Mbps < 3 Mbps	0	0	3	4.8
3 Mbps < 4 Mbps	0	0	4	6.5
4 Mbps < 5 Mbps	0	0	5	8.1
> 5 Mbps	0	0	17	27.5

Network Throughput Demands

Notice that out of the 62 traffic demands, 35 of the traffic demands exceed the 1 Mbps threshold. This accounts for 85.6% of the traffic that exceed the threshold. By definition, this course would be rated as **Network Intensive since it occurs over 50% of the time**. An example of a network intensive application can be an Adobe Flash Player with large size data, which this example shows above 20 MB using bulk download.

Remediation

If an application is rated as network intensive, the following options can be considered:

- The application is network intensive, but found to have a relatively minimal impact to the UTN.
 - The course can be posted on AgLearn, but the user performance might not be acceptable.
 - Or, the course can be redeveloped and reassessed.
- The application is network intensive, and found to have an impact to the UTN.
 - The TRB may need to consider whether to permit the release of the course.
 - Or, the course can be redeveloped and reassessed.

Points of Contact

Further questions for NIA guidelines can be directed to the NTSO staff listed below.

- John Velasquez
john.velasquez@ocio.usda.gov
970-295-5318
- Eun Ae Hwang
eun.hwang@ocio.usda.gov
970-295-5257

Note: The NTSO staffs are currently working on major modification to the criteria and adjusting guidelines accordingly for the NIA.