MMC: Content Server Solutions

Development and Consulting Proposal
Andrew Chemey
Updated: November 14th, 2016
# Table of Contents

1: PURPOSE ................................................................................................................................. 4

OVERVIEW ...................................................................................................................................... 4

Objective ......................................................................................................................................... 4

2: BACKGROUND ............................................................................................................................ 5

LMS-STANDARDS OVERVIEW ......................................................................................................... 5

AICC LMS STANDARD ..................................................................................................................... 5
SCORM LMS STANDARD .................................................................................................................. 5

CROSS-DOMAIN ENVIRONMENTS .................................................................................................. 5

LMS IMPLEMENTATION .................................................................................................................. 6

SINGLE-DOMAIN ENVIRONMENT .................................................................................................... 6
CROSS-DOMAIN ENVIRONMENT ..................................................................................................... 6

3: EXPAND MICHAEL MANAGEMENT OFFERINGS ........................................................................ 7

FEATURES ......................................................................................................................................... 7

 Migirate to HTML-based Content .................................................................................................. 7
  Current Functionality ..................................................................................................................... 7
  Proposed Solution ........................................................................................................................ 7
 Validate an MMC Learner to Prevent Unauthorized Access ......................................................... 8
  Current Functionality ..................................................................................................................... 8
  Proposed Solution ........................................................................................................................ 8
 Redirect MMC Learners to an MMC-hosted Page at the End of Lessons ...................................... 9
  Current Functionality ..................................................................................................................... 9
  Proposed Solution ........................................................................................................................ 9
 Redirect MMC Learners to an MMC-hosted Page at the End of Quizzes ...................................... 9
  Current Functionality ..................................................................................................................... 10
  Proposed Solution ........................................................................................................................ 10
 Support "Resume" Functionality for MMC Learners ...................................................................... 10
  Current Functionality ..................................................................................................................... 11
  Proposed Solution ........................................................................................................................ 11
  Back-end modifications ................................................................................................................. 11
  Captivate / Launch File Modifications ......................................................................................... 12
 Support Third-Party LMS Integrations .......................................................................................... 13
  Current Functionality ..................................................................................................................... 13
  Proposed Solution ........................................................................................................................ 13
  LMS Standards ............................................................................................................................ 13

4: REVIEW ......................................................................................................................................... 16

CAPTIVATE DEVELOPMENT ......................................................................................................... 16

HTML LAUNCH FILES .................................................................................................................... 16

SERVER-SIDE CHANGES ............................................................................................................... 16

Database Changes .......................................................................................................................... 16
ASP Files ......................................................................................................................................... 16

AICC/SCORM IMPORT FILES ......................................................................................................... 16

AICC .................................................................................................................................................. 16
1: Purpose

What needs to be accomplished?

Overview

Michael Management Corporation (MMC) utilizes their own content servers for hosting and launching Flash-based Captivate content (content) they have developed. They provide their own LMS-like solution (MLS) for access and tracking progress of learners. They would like to expand their current offerings:

1. Migrate to HTML-based content
2. Validate an MMC learner to prevent unauthorized access
3. Redirect MMC learners to an MMC-hosted page at the end of lessons
4. Redirect MMC learners to an MMC-hosted page at the end of quizzes
5. Support “resume” functionality for MMC learners
6. Support third-party LMS integrations
   a. Content hosted (and licenses managed) by MMC
   b. Content hosted by third-party LMS (licenses managed by MMC)
   c. Content hosted by third-party LMS (Sessions/learners managed by third-party LMS)

A Captivate Widget (widget) has been created to facilitate the communication between the content and the MLS. In addition, a series of ASP pages have been created to handle communication from the content.

Objective

The purpose of this document is to propose solutions to allow MMC to expand their offerings for their MLS users, as well as support third-party LMS solutions.
2: Background
LMS Standards and Environmental Information

LMS-Standards Overview

There are two primary LMS standards:

- AICC
- SCORM

AICC Standard

AICC stands for Aviation Industry CBT (Computer-Based Training) Committee. It is the original standard for LMS communication. It uses an HTTP-POST command for sending data fields to/from an LMS. It is also the most “domain-friendly” standard, to help support content on one domain, and the LMS on another.

SCORM Standard

SCORM stands for Sharable Content Object Reference Model. It uses JavaScript function calls and has the same default data elements as AICC. It tends not be very domain friendly and unless supported by an LMS, needs to have the content and LMS on the same domain.

Cross-Domain Environments

Cross-domain is the transfer of information/data from one domain to another. There are security restrictions or issues with cross-domain communication. In general, all browsers do not allow one domain to communicate with another domain. The intent is to prevent nefarious communication from occurring. Since browsers tend to prevent all cross-domain communication from occurring, any attempts to use HTML/JavaScript will not work to post data cross-domain.

However, just because a browser prevents cross-domain communication, it doesn't prevent other solutions from allowing it under various conditions. Flash allows cross-domain communication, under certain conditions. Server-side scripts such as ASP, PHP and .NET can also allow cross-domain communication. JavaScript technologies such as jQuery can also communicate cross-domain via an HTTP-GET and HTTP-POST. However, being able to host content in a frameset and communicate between frames hosted in different domains is NOT supported under any environment or technology.
LMS Implementation

MMC hosts their content and their own LMS on the same domain. This constitutes a single-domain environment.

Single-Domain Environment

The MMC LMS environment has both the LMS and the content hosted within the same domain. This environment allows any content to communicate with MMC without any special considerations.

NOTE: The MMC LMS/Content environment does NOT use the AICC/SCORM standards for communication. It uses a Captivate widget which communicates with the MMC Server via a number of ASP calls.

Cross-Domain Environment

Communication with any third-party LMS, typically means that the content will be hosted on one domain (e.g. www.michaelmanagement.com) and the LMS is hosted on another domain (e.g. client.lms.com). This constitutes a cross-domain environment, which has limitations/restrictions for communication.
3: Expand Michael Management Offerings

Michael Management LMS Experience

Today, MMC provides a great integrated experience of managing users training and content. This solution has worked for a number of years. MMC would like to expand their offerings to provide additional features.

Features

Here are some of the features that MMC would like to provide with their existing LMS:

Migrate to HTML-based content

The current content provided by MMC is Captivate content, published to the SWF format. To be able to maintain relevancy as the Flash Player becomes less supported; and to expand into mobile markets, MMC would like to publish their content to HTML5. After substantial testing, it has been determined that Captivate can publish high-quality content to HTML5. However, the solution to-date is completely Flash-based.

Current Functionality

There are two ASP files and a separate flashWrapper used to launch existing Flash-based content:

- start.asp
- getLesson.asp
- flashWrapper.swf

Start.asp file is used to ensure someone has logged in and then builds html/JavaScript to launch ONLY Flash-based files. In the current model, it launches flashWrapper.swf and passes to that file some parameters. flashWrapper.swf uses those parameters to call getLesson.asp to determine the location/name of the actual course launch file. Part of the purpose for getLesson.asp is to obfuscate the launch file, so someone wouldn’t easily be able to wean the URL directly to the content files and launch/view the training without being a paid customer or being logged-in.

Proposed Solution

To support HTML-based content/tracking, a new start.asp is required (called startHybrid). This file would handle the logic previously performed by getLesson.asp AND the flashWrapper. This solution exposes the actual URL to the course files, if the user views the browser source. This is an issue for the existing solution, but can be resolved by validating the MMC learner in the content.
Validate an MMC learner to prevent unauthorized access

With the current solution, anyone that is able to access the URL’s to the content, can launch and take the content without any issues.

Current Functionality

Even with the current implemented solution, the URL to the content files are accessible, if users really wanted to get them. Pressing F12 in any browser will bring up debugging tools which show every URL that is called. Here’s an example:

<table>
<thead>
<tr>
<th>URLs (hosted on: <a href="https://www.michaelmanagement.com">https://www.michaelmanagement.com</a>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/start.asp?lid=con102_01&amp;pid=con102</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/index.html?lid=con102_01&amp;pid=con102&amp;cid=67167</td>
</tr>
<tr>
<td>/favicon.ico</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/htmlimages/loader.gif</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/js/CPXHRLoader.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/css/CPLibraryAll.css</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/js/jquery-1.11.3.min.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/js/CPM.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/playbar/playbarScript.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/htmlimages/correct_answer_normal.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/htmlimages/correct_answer_small.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/htmlimages/correct_question_normal.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/htmlimages/correct_question_small.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/playbar/PlaybarIcons/AudioOff.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/playbar/PlaybarIcons/AudioOn.png</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/ar/66157.mp3</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/ar/Mouse.mp3</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/wr/w_69440/Widget_69440.htm</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/wr/10990.mp3</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/wr/69434/js/jquery-3.1.1.min.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/wr/69434/js/widgetMMC_HybridLesson.js.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/js/jquery-1.11.3.min.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/assets/js/OpenAjaxManagedHub-all.js</td>
</tr>
<tr>
<td>/files/training/con102/con102_01/wr/w_69440/js/widgetMMC_HybridLesson.js.js</td>
</tr>
</tbody>
</table>

This is only a small snippet, and it’s obvious that every single file GET/POST is available from the browser tools. To help prevent unauthorized access to the content, simply obfuscating the URL that is available if the user clicks “view source” is not enough.

Proposed Solution

The content should make a call to MMC to validate the session. This can be accomplished by having the widget should call validate.asp file on the server. Validate.asp should determine, based on the parameters passed (PID, CID, LID), that the user is authorized to view the content.
Redirect MMC learners to an MMC-hosted page at the end of lessons

It is recommended to change the process used to redirect users to a completion page at the end of the content.

Current Functionality

Currently, the Captivate content performs a call to logComp.asp. This is set in the Quiz Preferences of Adobe Captivate:

Proposed Solution

Since the Widget handles all communication with the MMC server, it's recommended to have the widget redirect to MMC server at the end, too.

Redirect MMC learners to an MMC-hosted page at the end of quizzes

The experience for MMC learners is different for lessons than quizzes (exams). It is recommended to keep the experience consistent.
**Current Functionality**

Currently, at the end of lessons, the content closes and the learner is redirected to an MMC web page that shows the number of points received and they can view additional information:

![Lesson Complete](image)

However, quizzes do not provide this same experience. Quizzes are spawned into a new window. At the end of the quiz, the new window is closed. It would be nice to provide a consistent experience, regardless of the content launched.

**Proposed Solution**

Since the Widget handles all communication with the MMC server, it’s recommended to have the widget redirect to MMC server at the end of quizzes, too.

**Support “resume” functionality for MMC learners**

Resume provides the ability to track learners progress, such that each time they launch the content (lesson or quiz), they return to where they left off and it remembers how they answered questions previously and the number of times they attempted the quiz, etc.
Current Functionality

Both lessons and quizzes must be completed in a single session for any tracking to occur. I believe number of attempts may be tracked on the MMC server, however, time spent, completion and scores are only tracked when the content is completed in a single session.

Each time a lesson or quiz is launched, the learner starts from the beginning. When a quiz is launched, a learner seemingly has the ability to see/answer questions and before reaching the end, they can quit. A score won’t be recorded and they can relaunch, with the ability to re-answer questions until they receive a passing score.

Proposed Solution

Adobe Captivate provides the ability to resume, however it requires publishing to and using an AICC/SCORM LMS. The proposed solution is to create a wrapper which makes the Captivate file believe it is interacting with an AICC/SCORM LMS to provide the necessary resume functionality.

Back-end modifications

Regardless of whether AICC or SCORM is used, the MMC back-end database needs to be modified to support the resume fields. A table needs to be created/modified with these fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product ID</td>
<td>VarChar (128)</td>
<td>Standard field already used which contains the Product ID</td>
</tr>
<tr>
<td>Course ID</td>
<td>VarChar (128)</td>
<td>Standard field already used which contains the Course ID</td>
</tr>
<tr>
<td>Lesson ID</td>
<td>VarChar (128)</td>
<td>Standard field already used which contains the Lesson ID</td>
</tr>
<tr>
<td>Location</td>
<td>VarChar (255)</td>
<td>Location of the slide where the learner was last at. This may not be required, but it’s simple enough to use/store</td>
</tr>
<tr>
<td>Status</td>
<td>VarChar (25)</td>
<td>The status of the content (e.g. incomplete, completed, passed). This may not be required, but it’s simple to do so</td>
</tr>
<tr>
<td>Score</td>
<td>Integer</td>
<td>The current score of the content. This may not be required, but it’s simple enough to use/store</td>
</tr>
<tr>
<td>Time</td>
<td>VarChar (25)</td>
<td>The amount of time the learner has spent in the content.</td>
</tr>
<tr>
<td>Resume Data</td>
<td>Blob</td>
<td>Additional data Captivate uses to know where to resume the learner in a Captivate file, including quiz attempts and questions answered</td>
</tr>
</tbody>
</table>

Additional ASP files will either need to be created or modified to help support the resume functionality. However, these ASP changes will be based on which LMS standard is used.
Captivate / Launch File Modifications

A SCORM LMS provides a JavaScript API, which Captivate would interact with, when published with SCORM tracking enabled. The proposed solution is to provide this SCORM API with data from MMC which would support resume. Start.asp would be modified to read from the database to wean the resume data information for the learner. The HTML code generated by Start.asp would include this data AND basic JS to make Captivate believe it is communicating with an LMS:

```html
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv='Content-Type' content='text/html; charset=utf-8'>
  <title>Online SAP Training by Michael Management</title>
  <script type='text/javascript' language='JavaScript'>
    var API = new Object();
    API.LMSInitialize = function() {
      return 'true';
    }
    API.LMSGetValue = function(property_str) {
      switch (property_str) {
        case 'cmi.core._children':
          return 'lesson_location, score, lesson_status, session_time';
        case 'cmi.core.student_id':
          return 'AChemey';
        case 'cmi.core.student_name':
          return 'Chemey, Andrew';
        case 'cmi.core.lesson_location':
          return '3';
        case 'cmi.core.score._children':
          return 'raw, min, max';
        case 'cmi.core.lesson_status':
          return 'incomplete';
        case 'cmi.core.session_time':
          return '05:18:24';
        case 'cmi.suspend_data':
          return 'dk38224kd@dsdg11295';
      }
    }
  </script>
</head>
<frameset cols='*'>
  <frame src='index_lms.html'>
</frameset>
</html>
```

The simple snippet of HTML code provided above, will actually allow a SCORM-packaged Captivate content to believe it is communicating with a SCORM-compliant LMS and would resume a learner to where they left off.
Storing the data could also be relatively simple. It could be accomplished using the widget, or pure JavaScript. To be consistent, I would recommend using the SCORM/JavaScript mechanism to store the data locally, and the widget to send the data once, when the last slide is reached. To handle cases where the learner exits BEFORE the last slide is reached, the HTML file should send data on browser close, if the data hasn’t already been sent.

**PROS**
The PRO’s of using this mechanism, is that it is extremely east to set up and use. It could possibly be accomplished in a day (including all database, ASP and JS development).

**CONS**
Up until this point, all Captivate files have been published without any LMS tracking enabled. The Captivate content would have to be published for SCORM., to support this mechanism. That by itself is menial and irrelevant. However, a separate goal for the overall project, and discussed later in this document, is to integrate with third-party LMS’s, mainly in cross-domain environments. Since we don’t want to have to republish multiple times, we will have to use a SCORM-to-AICC wrapper to publish all content as SCORM. Any Third Party AICC integrations will have to use a wrapper.

**Support Third-Party LMS Integrations**

MMC would like to provide a solution to their customers that have their own LMS, but would like to purchase MMC licensed content. There are three use-cases to support Third-Party integrations:

1. Content hosted and licenses managed by MMC
2. Content hosted by third-party LMS, but licenses managed by MMC
3. Content hosted and Licenses managed by third-party LMS

Obviously, the primary use-case that we would like to support is content hosted and licenses managed by MMC. However, we should at least discuss the other options, in case those scenarios arise.

**Current Functionality**

There isn’t any current support for third-party LMS integrations.

**Proposed Solution**

The proposed description is easy – provide support for third-party LMS integrations. However, it’s a complicated solution. The primary focus is to ensure that customers have a seamless integration when they are within their defined license agreements; and cannot exceed or workaround their defined licenses.

**LMS Standards**

As described before, there are two LMS standards: AICC and SCORM. AICC is more cross-domain friendly and that plays a significant role in how the third-party solutions work.

**AICC Solution**

The AICC solution may be relatively easy. Here’s how it would work for the three scenarios
Content hosted and licenses managed by MMC
When an AICC-compliant LMS launches content, it passes at least two parameters in the URL: AICC_SID and AICC_URL. We would have the LMS add three more parameters to the URL to include: PID, LID and CCODE.

PID: The Product ID as it exists today in the product.

LID: The Lesson ID as it exists today in the product

CCODE: A new, unique Company Code. It represents an individual company using third-party LMS integration. This may be something like GM123

We will build a launcher (or wrapper) file that is launched by the LMS. This launcher file will communicate with the LMS to wean the User ID (which we’ll call CID) and pass these four parameters to the content. When the content is launched, the widget will be modified to send these four parameters to validate.asp. Validate.asp will be modified to look for these 4 parameters (NOTE: The validate.asp was described earlier as expecting 3 parameters – PID, LID and CID). When validate.asp receives 4 parameters, it will perform a lookup on the CCODE (Company Code) to determine how many licenses that company has purchased. Then it will store the User ID (CID) into its database and determine how many unique users have launched the content to-date. If this user exceeds their allotted limit, the widget will perform a redirect to a custom page (on MMC?) that indicates their licenses have been exceeded and they need to contact their Administrator.

To support this, the back-end database needs to change, validate.asp has to change and the widget has to change. In addition, custom AICC Course Structure files will be uniquely generated for each customer, which they will use to import the content into the LMS.

Content hosted by third-party LMS, but licenses managed by MMC
The solution for third-party LMS integrations where the content is hosted by the LMS – the solution is exactly the same as when the content is hosted by MMC. The PID, LID and CCODE will be part of the AICC Course Structure files, which the LMS administrator uses to import into their LMS. The widget will still reach out to Validate.asp hosted on MMC and will redirect to a custom page (on MMC?) if the license has been exceeded. The outstanding question is what to do, if an error occurred (e.g. validate.asp could not be reached). In that case, we might want to create a custom error page which ships with the product and is used as the redirect page.

Content hosted and Licenses managed by third-party LMS
I’m not sure I have a reasonable solution for content hosted AND licenses managed by the third-party LMS vendor. The suggestions I have would be a periodic check in the LMS Database for the number of users that have launched the content; a separate report that would be generated; or a separate database that is installed on the third-party LMS vendor’s servers which handles the license count.

SCORM Solution
To support a SCORM solution, we need to perform more server-to-server calls AND change the end-user experience slightly.

Content hosted and licenses managed by MMC
This is a cross-domain environment. The LMS is hosted on one domain, and the content is hosted on MMC. Just to get the content to launch and communicate with the LMS, an LMS launch file
needs to be created which will communicate with the LMS using the SCORM standard. Then the LMS Launch file would need to call an ASP file on MMC, which is a Content Launch file into a new browser window. The Content Launch file creates the SCORM-API which the content will communicate to. When the Content is closed; or reaches the last page – the content needs to call an MMC ASP to store the data in the database, temporarily. The LMS Launch Window, which is constantly listening for the Content Window to close, then calls an MMC-hosted ASP file to wean the data from the MMC database. The LMS Launch file updates the LMS with the data and then closes the browser window.

This solution seems easy – but it’s very timing-centric. Hopefully everything is communicated in a reasonable time-frame.

What was missing from this process is how to handle the licenses. The content would have to receive the same 4 parameters as described in the AICC Solution section (PID, LID, CCODE, CID).

Content hosted by third-party LMS, but licenses managed by MMC
This solution should be relatively straight-forward. The LMS would launch the content using the three parameters specified earlier (PID, LID, CCODE). The CID would be weaned from the User ID (from the LMS). These four parameters would be passed by the widget to the validate.asp. Any errors reaching the validate.asp or receiving a valid response, would redirect to an HTML file with the content. License issues would be re-directed either to an HTML file with the content or on MMC.

Content hosted and Licenses managed by third-party LMS
I’m not sure I have a reasonable solution for content hosted AND licenses managed by the third-party LMS vendor. The suggestions I have would be a periodic check in the LMS Database for the number of users that have launched the content; a separate report that would be generated; or a separate database that is installed on the third-party LMS vendor's servers which handles the license count.
4: Review

Requirements for a successful deployment

The key to a successful deployment, is to make all of this work with a single set of published Captivate files (with a single widget). We will have to develop multiple HTML launch files, ASP pages created/modified, database changes, handle client management (to create customer codes, license models, etc.) and create the AICC/SCORM import files.

Captivate Development

- Describe the changes to the Captivate Files
- Describe the changes to the Captivate Widget

HTML Launch Files

Describe the various HTML Launch Files and how / when they are used

Server-side Changes

This section describes the server-side changes required

Database Changes

DB Changes

ASP Files

ASP Files (new/modified)

AICC/SCORM Import Files

AICC

SCORM