

Matt Perkins

<http://www.nudoru.com>

kheavy@nudoru.com

WHAT IS SCORM?

AND

HOW CAN YOU USE IT WITH FLASH?

What is SCORM

Several years ago, the White House Office of Technology, the Department of Defense and the Department of Labor launched the **Advanced Distributed Learning Initiative (ADL)** in the United States. Industry and education partners, as well as various entities in other countries also joined the effort. One of the first ADL projects was for a practical profile of existing specifications and standards for content. A profile is a document that specifies a particular interpretation of a standard or specification. In the process, some gaps had to be filled in, and the end result was a set of "books", each describing a different aspect of the solution. The result was called the **Shareable Content Object Reference Model, or SCORM**. The SCORM was born to take the best from the early efforts, specifications and standards, and achieve the goals of durability, portability, reusability, interoperability and accessibility for content.

Source: Claude Ostyn, In the Eye of the SCORM

What does SCORM Stand For?

- Sharable
- Content
- Object
- Reference
- Model

A single SCORM “course” is called a **SCO**

SCORM Versions

SCORM 1.2

The first "real" release of the SCORM was SCORM 1.2. This was the first version for which a test suite was available, and thus the first version for which conformance could be verified. SCORM 1.2 proved that content can be made portable and interoperable.

SCORM 2004 (or 1.3)

SCORM 2004 improves significantly on SCORM 1.2, by eliminating even more ambiguities in the specification, and by making SCORM conformant with robust [...] standards. [...] Besides improving on SCORM 1.2, SCORM 2004 also adds optional features for sequencing and navigation.

The addition of sequencing is a major functional milestone. SCORM 1.2 was all about making content portable, but left it to the learner to choose which part of the content to run. SCORM 2004 adds the ability to deliver activity-centered content packages that support guided or adaptive sequencing behavior.

Source: Claude Ostyn, In the Eye of the SCORM

SCORM 1.2 – the SCO/course controls the sequencing of lessons through it's own custom logic.

SCORM 2004 – the LMS controls the sequencing of lessons/SCOs.

1.2 vs 2004 – My Take

- Unless you're taking advantage of the sequencing and navigation features that 2004 provides, use 1.2
- If you know that you need to use 2004 and can support yourself, use it

Limited tool support for 2004 manifests = hand coding XML to spec = **pain**

Manifest editors are available. I don't have much experience with them. However, they are not user friendly.

Levels of Authoring Tools

- **Easy** – no problems for designers or SMEs to use
- **Medium** – ok for “technically minded” designers
- **Hard** – only for programmers

SME	X		
Designer	X	X	
Developer		X	X
	Easy	Medium	Hard

Easy



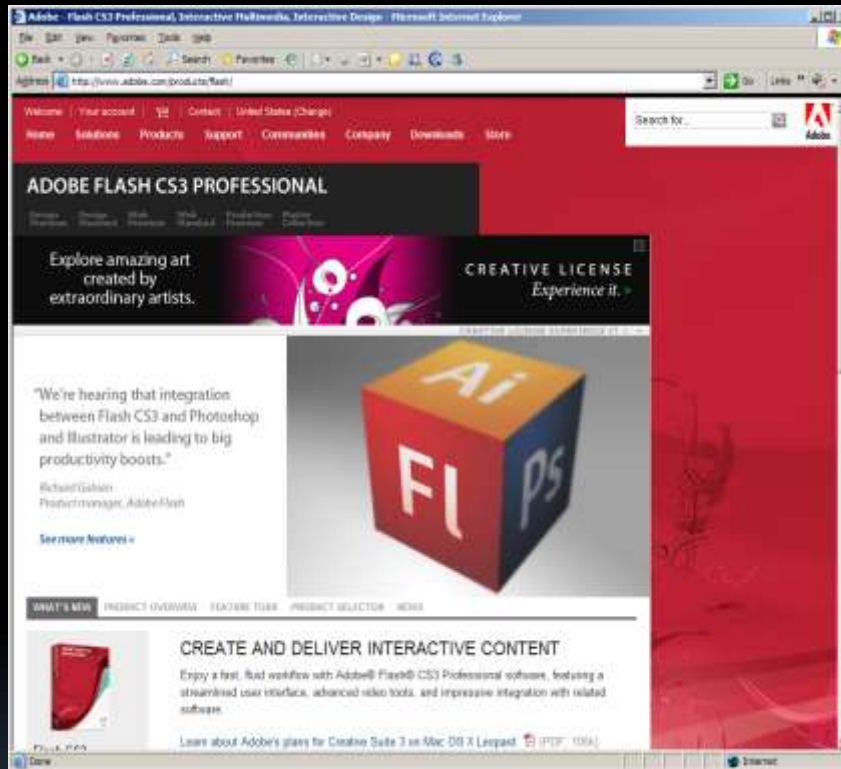
- SCORM tracking is handled automatically
- Very little consideration past “passing score”

Medium



- The technical details of talking to the LMS is handled for you
- You need to handle lesson status, passing score and a few other things

Hard



- You're own your own
- Need to know how to talk to the LMS
- Need to know variable names and acceptable values

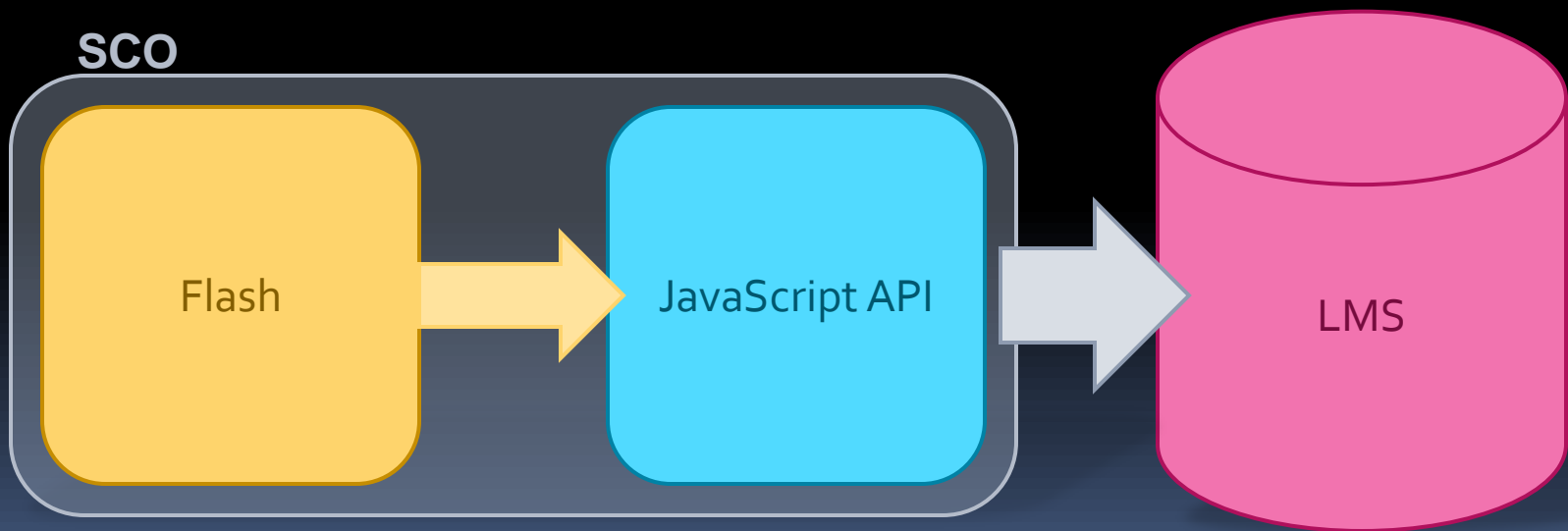


You already know about Lectora, so ...

LET'S FOCUS ON THE HARD WAY

Moving Parts

The Flash* course must contain code that talks to JavaScript functions in the HTML. The JavaScript talks to the SCORM API from the LMS (it's in a hidden frame).



*Flash in this case could be any type of learning content: Articulate, Lectora, etc. This is the way all SCORM communication happens.

Making it easier

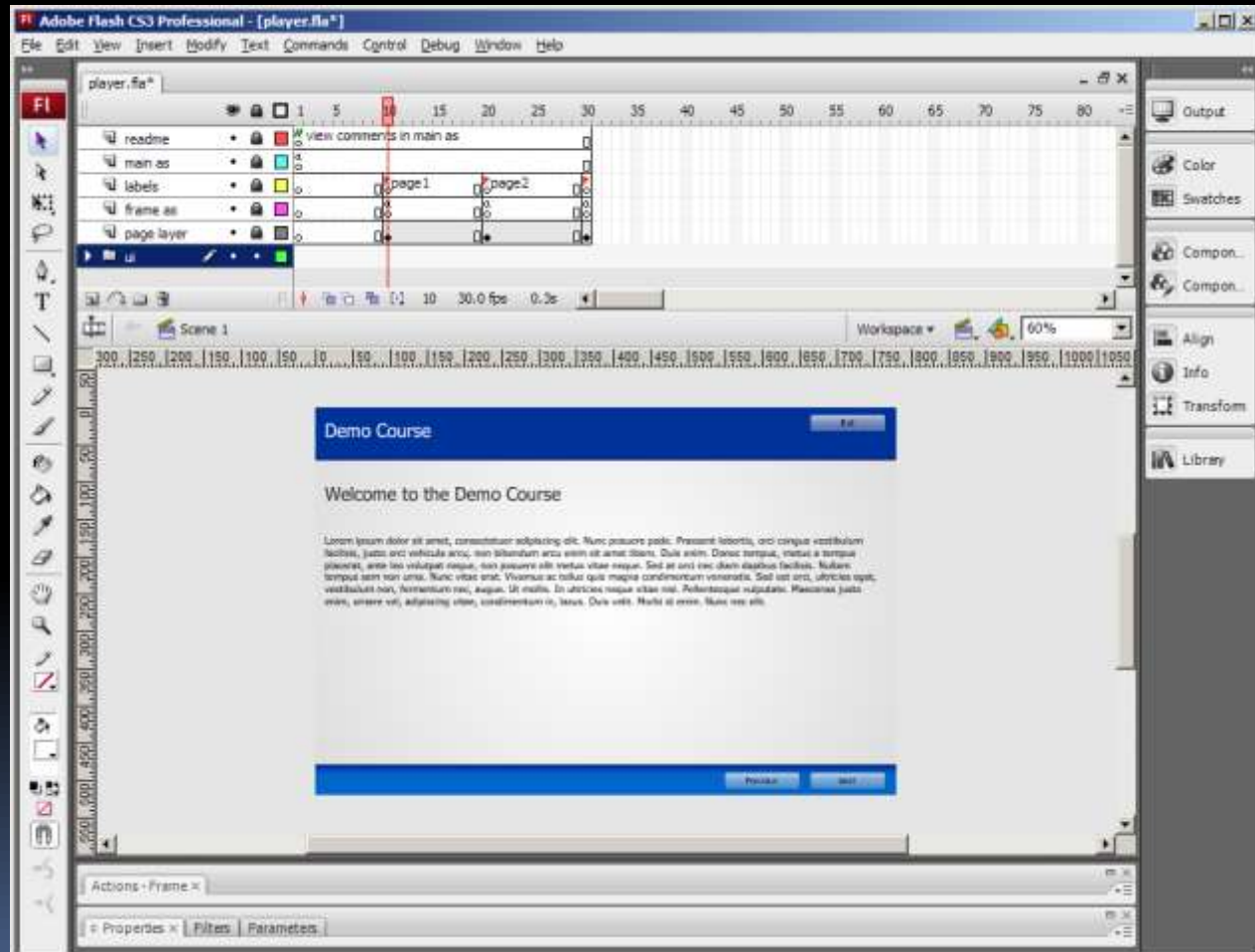
- Pipwerks SCORM Wrapper for Flash
 - Replaces the ADL's JavaScript API
 - AS3 SCORM class that works with the JavaScript API
 - Works with 1.2 or 2004
 - Tested on many popular LMS systems

<http://pipwerks.com/lab/scorm/wrapper/flash/index.html>

I Made It Even Easier

- LMSData class wraps the Pipwerks wrapper
- And SCO template – includes AS classes, JavaScript, HTML, and a sample manifest

Example – Using LMSData With a Timeline Based WBT



First Frame

- Establish variable and make connection to the LMS

```
// create the LMS object
var LMSConnection:LMSData = new LMSData();

// initialize the LMS connection
if (LMSConnection.initialize()) {
    trace("Successfully connected to the LMS!");
    // get the bookmark, but validate it first
    var lastLoc:int = int(LMSConnection.lastLocation);
    if(lastLoc > 9) gotoAndStop(lastLoc);
    // the bookmark wasn't valid, so just start at the first page
    else play();
} else {
    trace("There was a problem connecting to the LMS!");
    play();
}
```

On Each Page

- Set a bookmark

```
// set the bookmark  
LMSCConnection.lastLocation = String(this.currentFrame);
```

Last Page

- Set a bookmark
- Set complete

```
// set the bookmark
LMSCConnection.lastLocation = String(this.currentFrame);

// we're done!
LMSCConnection.setComplete();
```

Exit Button

- Close the connection – and the window is closed automatically from the script

```
// do this when the exit button is clicked
exit_btn.addEventListener(MouseEvent.CLICK, onExitClick);
function onExitClick(e:Event):void {
    LMSConnection.exitCourse();
}
```



Flash WBT “Framework”

- Reusable templated system for quick creating Flash WBTs
- Templates != page layouts (but could)
- Heavily script based
- Over come the limitations of the timeline

Timeline vs Framework

Timeline	Framework
Pages added on the timeline	Pages added as variables
Add/delete/move pages by painstaking frame copy/pastes	Copy/paste lines of text
Content must be in the FLA file = large file sizes	Content may be in XML or external SWFs = multiple small files
"One off" or custom WBT	Easily repurpose and reuse code across many WBTs
Easy to do for a novice Flasher	Easy to do for an intermediate Flasher
Quick to create, but same work is repeated with each WBT	Long time to create the first one, but work is reused and maintenance time is saved



Framework demo

- Example: Framework example

<http://udon.nudoru.com/2008/05/20/creating-a-flash-wbt-framework-part-1/>

- Advanced: inFlite / Scientia / Ramen

<http://udon.nudoru.com/ramen-player/>





For more information:
kheavy@nudoru.com

THAT'S ALL!