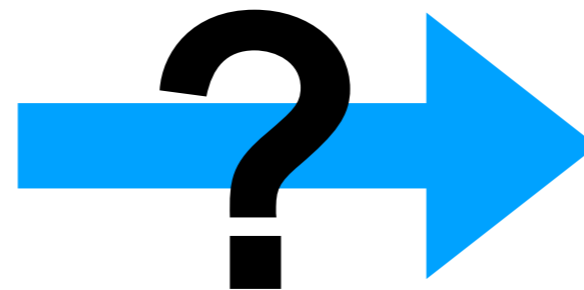


# SCORM to cmi5

An Implementor's Guide



## Bits To Convert

---

Manifest: What's inside?

Launch: Make it go!

Runtime: Tell me about it!

# What's a Manifest?

- ❑ Central component of content package
- ❑ Describes structure of material
- ❑ Indicates launchable components
- ❑ XML file



# Manifest Comparisons

	SCORM	cmi5
<b>Filename</b>	imsmanifest.xml	cmi5.xml
<b>Activity Tree</b>	SCO	AU
<b>Activities</b>	Activity Leaf Node	AU
<b>Containers in the Tree</b>	Aggregation	Block

**SCO = Sharable Content Object**

**AU = Assignable Unit**

# SCORM 1.2 Manifest

---

```
<?xml version="1.0" encoding="utf-8"?>
<manifest identifier="_62yRKuq1asJ_course_id" version="1.0">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
  </metadata>
  <organizations default="Simple_SCORM_Packager_Tutorial_ORG">
    <organization identifier="Simple_SCORM_Packager_Tutorial_ORG">
      <title>Simple SCORM Packager Tutorial</title>
      <item identifier="Simple_SCORM_Packager_Tutorial_SCO" isvisible="true"
        identifierref="__62yRKuq1asJ_course_id_RES">
        <title>Simple SCORM Packager Tutorial</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="__62yRKuq1asJ_course_id_RES" type="webcontent"
      href="index_lms.html" adlcp:scormtype="sco">
      <file href="story_content/thumbnail.jpg"/>
    </resource>
  </resources>
</manifest>
```

# Convert to cmi5!

```
<?xml version="1.0" encoding="utf-8"?>
<manifest identifier="__62yRKuq1asJ_course_id" version="1.0">
  <metadata>
    <schema>ADL_SCORM</schema>
    <schemaversion>1.2</schemaversion>
  </metadata>
  <organizations default="Simple_SCORM_Packager_Tutorial_ORG">
    <organization identifier="Simple_SCORM_Packager_Tutorial_ORG">
      <title>Simple SCORM Packager Tutorial</title>
      <item identifier="Simple_SCORM_Packager_Tutorial_SCO" isvisible="true"
        identifierref="__62yRKuq1asJ_course_id_RES">
        <title>Simple SCORM Packager Tutorial</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="__62yRKuq1asJ_course_id_RES" type="webcontent"
      href="index_lms.html" adlcp:scormtype="sco">
      <file href="story_content/thumbnail.jpg"/>
    </resource>
  </resources>
</manifest>
```

**SCORM**

```
<?xml version="1.0" encoding="utf-8"?>
<courseStructure xmlns="https://w3id.org/xapi/profiles/cmi5/v1/CourseStructure.xsd">
  <course id="http://jcasolutions.com/courses/_62yRKuq1asJ_course_id">
    <title>
      <langstring lang="en-US">Simple SCORM Packager Tutorial</langstring>
    </title>
    <description>
      <langstring lang="en-US">Simple SCORM Packager Tutorial</langstring>
    </description>
  </course>
  <au id="http://jcasolutions.com/courses/_62yRKuq1asJ_course_id/activity1">
    <title>
      <langstring lang="en-US">Simple SCORM Packager Tutorial</langstring>
    </title>
    <description>
      <langstring lang="en-US">Simple SCORM Packager Tutorial</langstring>
    </description>
    <url>index_lms.html</url>
  </au>
</courseStructure>
```

**cmi5**

# Other Things To Convert

## Mastery score

### SCORM 1.2:

```
<adlcp:masteryscore>100</adlcp:masteryscore>
```

### SCORM 2004:

```
<imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
```

### cmi5:

```
<au ... masteryScore="1.0">
```

# Other Things To Convert

## Per-SCO/AU Parameters

### SCORM 1.2 and 2004:

`<item parameters="a=1&b=2">`

or

`<resource href="...?a=1&b=2">`

### cmi5:

`<au>`

`<launchParameters>{'a':1,'b':2}</launchParameters>`

`</au>`



# Impedance Mismatch

Things there aren't equivalents for:

- ✧ SCORM 1.2: None

- ✧ SCORM 2004:

  - ✧ Sequencing

  - ✧ Most parts of rollup

    - ✧ “moveOn” or bust



# Launch!

---



# Launch

## LMS Responsibilities

- ✦ Lots!

## AU (your content's) Responsibilities

- ✦ Get your auth token
- ✦ Only get it once!

Get the Launch Data

# Launch URL

## Query parameters

- ✦ actor
- ✦ activityId
- ✦ endpoint
- ✦ fetch
- ✦ registration

# Launch URL

```
http://www.enfixlp.com/ssptutorial/index_lms.html  
?endpoint=http://enfixlp.com/xapip/base/  
&fetch=http://enfixlp.com/xapip/authtoken?k=2390289x0  
&actor={"objectType": "Agent", "account":  
{"homePage": "http://enfixlp.com", "name": "gav"}}  
&registration=760e3480-ba55-4991-94b0-01820dbd23a2  
&activityId=http://jcasolutions.com/courses/_62yRKuq1asJ
```

# Launch URL Breakdown

`http://www.enfixlp.com/ssptutorial/index_lms.html`

`?endpoint=http://enfixlp.com/xapip/base/`

`&fetch=http://enfixlp.com/xapip/authtoken?k=2f39028d9`

`&actor={"objectType": "Agent", "account":`

`{"homePage": "http://enfixlp.com", "name": "gav"}`

`&registration=760e3480-ba55-4991-94b0-01820dbd23a2`

`&activityId=http://jcasolutions.com/_62yRKuq1asJ`

LRS URL

Auth URL

Learner

Registration  
ID in LMS

Course  
Activity ID

# Getting the Auth Token

HTTP POST to “fetch” URL

EX: <http://enfixlp.com/xapip/authtoken?k=2f39028d9>

JSON response with token:

```
{  
  "auth-token": "QWxhZGRpbjpvGcVulHNlc2FtZQ=="  
}
```

Don't call it again!

```
{  
  "error-code": "1",  
  "error-text": "The authorization token has already  
    been returned."  
}
```

# Launch Data

## LMS-provided

- ✳ Describes context information for student launch
- ✳ Defines custom launch parameters
- ✳ Tells course where to go after completion

## Stored as an xAPI Activity State

- ✳ Looked up by:
  - ✳ activityId: The AU ID
  - ✳ agent: The agent from the Launch URL
  - ✳ registration: The registration from the Launch URL
  - ✳ stateId: "LMS.LaunchData"



# Launch Data

---

```
{
  "contextTemplate": {
    "contextActivities": {
      "grouping": [
        {
          "objectType": "Activity",
          "id": "http://jcasolutions.com/courses/_62yRkuq1asJ/activity1",
          "definition": {
            "name": {
              "en-US": "Simple SCORM Packager Tutorial"
            },
            "description": {
              "en-US": "..."
            }
          }
        }
      ]
    },
    "extensions": {
      "https://w3id.org/xapi/cmi5/context/extensions/sessionid":
      "3738c07f-e8f2-46df-8d9e-11ccda3a191c"
    }
  },
  "launchMode": "Normal",
  "returnURL": "http://enfixlp.com/courses/",
  "launchMethod": "AnyWindow",
  "launchParameters": "Start=1&FastForward=On",
  "entitlementKey": {"courseStructure": "xyz-123-9999", "alternate": "abc-456-1111"},
  "moveOn": "CompletedOrPassed"
}
```

# Runtime

---



# SCORM Runtime Environment Commands

## SCORM 1.2

## SCORM 2004

LMSInitialize("")

Initialize("")

LMSSetValue(key, value)

SetValue(key, value)

LMSGetValue(key)

GetValue(key)

LMSFinish("")

Terminate("")

# xAPI Verbs

Theoretically, any Internationalized Resource Identifier (IRI)

In practice: Usually known, well-defined verbs in IRI form

For this talk, verbs defined in the SCORM xAPI Profile:

✱ initialized

✱ terminated

✱ scored

✱ completed

✱ etc.

# Runtime, Compared

SCORM 1.2	SCORM 2004	xAPI / cmi5 verbs
LMSInitialize("")	Initialize("")	"initialized"
LMSSetValue(key, value)	SetValue(key, value)	<i>(Mapped by key, <sup>[ ]</sup>HTTP POST/PUT)</i>
LMSGetValue(key)	GetValue(key)	<i>(Mapped by key, <sup>[ ]</sup>HTTP GET)</i>
LMSFinish("")	Terminate("")	"terminated"

# A cmi5 Session, Briefly

Step Number	Action
Prequel	Course is launched; launch data is gathered
0	Session opens: "initialized"
1 to (N-1)	<One or more xAPI statements>
N	Session closes: "terminated"

# Initialized

Simple, just use the initialized verb

<http://adlnet.gov/expapi/verbs/initialized>

Let's try it!

# Initialized

---

```
{
  "actor": {
    "objectType": "Agent",
    "name": "George Vilches",
    "account": {
      "homePage": "https://enfixlp.com/",
      "name": "gav"
    }
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/initialized",
    "display": {
      "en": "initialized"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://jcasolutions.com/courses/_62yRkuq1asJ/activity1",
    "definition": {
      "name": {
        "en-US": "Simple SCORM Packager Tutorial"
      },
      "description": {
        "en-US": "..."
      }
    }
  }
}
```



# Uh oh...

---

Name	×	Headers	Preview	Response	Cookies	Timing
<input type="checkbox"/> statements /tc/9PZB8BH29Y	1	Forbidden cmi5 allowed statement: session not active				
1 requests   621 B transferred		Line 1, Column 1				

Uh oh...

---

Valid xAPI  $\neq$  Valid cmi5

# cmi5 Statements

## “cmi5 defined”

Known quantities

## “cmi5 allowed”

Fits within the session

## “cmi5 not-allowed”

Everything else

# “cmi5 defined”

Defined in the cmi5 xAPI Profile

“context” requires several parts

- ✳ LMS.LaunchData
  - ✳ “contextTemplate”
  - ✳ Include everything in every statement.
- ✳ Must have cmi5 category activity
  - ✳ “id”:  
"https://w3id.org/xapi/cmi5/context/categories/cmi5"

```
{
  "contextTemplate": {
    "contextActivities": {
      "grouping": [
        {
          "objectType": "Activity",
          "id": "http://jcasolutions.com/courses/_62yRKuq1asJ/activity1",
          "definition": {
            "name": {
              "en-US": "Simple SCORM Packager Tutorial"
            },
            "description": {
              "en-US": "..."
            }
          }
        }
      ]
    },
    "extensions": {
      "https://w3id.org/xapi/cmi5/context/extensions/sessionid":
      "3738c07f-e8f2-46df-8d9e-11ccda3a191c"
    }
  },
}
```

# “cmi5 defined” (cont’d)

## Actor

✳ Must be objectType: “Agent”

✳ Must be “account” type

## Limited verb choices

“id” required

“timestamp” required

# “cmi5 allowed”

Only usable within the session

Between initialized and terminated

All other valid xAPI statements

“id” still required

“timestamp” still required

Recommended: xAPI SCORM Profile

# Initialized, Take 2

```
{
  "id": "6f8e1135-c9fe-4e02-ae1f-a0f7468745eb",
  "actor": {
    "objectType": "Agent",
    "name": "George Vilches",
    "account": {
      "homePage": "https://enfixlp.com/",
      "name": "gav"
    }
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/initialized",
    "display": {
      "en": "initialized"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://jcasolutions.com/courses/_62yRKuq1asJ/activity1",
    "definition": {
      "name": {
        "en-US": "Simple SCORM Packager Tutorial"
      },
      "description": {
        "en-US": "..."
      }
    }
  },
  "context": {
    "registration": "4eb0e063-669b-479a-86b3-f9be9ac88a1d",
    "contextActivities": {
      "grouping": [
        {
          "objectType": "Activity",
```

```
      "id": "http://jcasolutions.com/courses/_62yRKuq1asJ/activity1",
      "definition": {
        "name": {
          "en-US": "Simple SCORM Packager Tutorial"
        },
        "description": {
          "en-US": "..."
        }
      }
    },
    "category": [
      {
        "objectType": "Activity",
        "id": "https://w3id.org/xapi/cmi5/context/categories/cmi5"
      }
    ],
    "extensions": {
      "https://w3id.org/xapi/cmi5/context/extensions/sessionid":
      "c7b6f0a9-482c-4c03-acc1-548289126963"
    },
    "timestamp": "2016-06-09T15:34:26.887Z"
  }
}
```

Red = cmi5 required components

## User Statuses

---

Passed / Failed: exactly one,  
exactly once

Completed: exactly once



# Passed / Failed

## SCORM 1.2:

LMSSetValue("cmi.core.lesson\_status", "passed")

## SCORM 2004:

SetValue("cmi.success\_status", "passed")

## cmi5

- ✳ "result" key required
- ✳ "score" optional
- ✳ "duration" required

# Passed / Failed

```
{
  "id": "6f8e1135-c9fe-4e02-ae1f-a0f7468745eb",
  "actor": {
    "objectType": "Agent",
    "name": "George Vilches",
    "account": {
      "homePage": "https://enfixlp.com/",
      "name": "gav"
    }
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/passed",
    "display": {
      "en": "initialized"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://jcasolutions.com/courses/_62yRkuq1asJ/activity1",
    "definition": {
      "name": {
        "en-US": "Simple SCORM Packager Tutorial"
      },
      "description": {
        "en-US": "..."
      }
    }
  },
  "result": {
    "success": true,
    "duration": "PT9.55S"
  },
  "context": {
    "registration": "4eb0e063-669b-479a-86b3-f9be9ac88a1d",
    "contextActivities": {
      "grouping": [
        {
          "objectType": "Activity",
          "id": "http://jcasolutions.com/courses/_62yRkuq1asJ/activity1",
          "definition": {
            "name": {
              "en-US": "Simple SCORM Packager Tutorial"
            },
            "description": {
              "en-US": "..."
            }
          }
        }
      ],
      "category": [
        {
          "objectType": "Activity",
          "id": "https://w3id.org/xapi/cmi5/context/categories/cmi5"
        }
      ],
      "extensions": {
        "https://w3id.org/xapi/cmi5/context/extensions/sessionid": "c7b6f0a9-482c-4c03-acc1-548289126963"
      },
      "timestamp": "2016-06-09T15:34:26.887Z"
    }
  }
}
```

# Completed

## SCORM 1.2:

LMSSetValue("cmi.core.lesson\_status", "completed")

## SCORM 2004:

SetValue("cmi.completion\_status", "completed")

## xAPI

 "result" key required

- ▶ "score" optional
- ▶ "duration" required

# Completed

```
{
  "id": "6f8e1135-c9fe-4e02-ae1f-a0f7468745eb",
  "actor": {
    "objectType": "Agent",
    "name": "George Vilches",
    "account": {
      "homePage": "https://enfixlp.com/",
      "name": "gav"
    }
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": {
      "en": "initialized"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://jcasolutions.com/courses/_62yRKuq1asJ/activity1",
    "definition": {
      "name": {
        "en-US": "Simple SCORM Packager Tutorial"
      },
      "description": {
        "en-US": "..."
      }
    }
  },
  "result": {
    "completion": true,
    "duration": "PT9.55S"
  },
  "context": {
    "registration": "4eb0e063-669b-479a-86b3-f9be9ac88a1d",
    "contextActivities": {
      "grouping": [
        {
          "objectType": "Activity",
          "id": "http://jcasolutions.com/courses/_62yRKuq1asJ/activity1",
          "definition": {
            "name": {
              "en-US": "Simple SCORM Packager Tutorial"
            },
            "description": {
              "en-US": "..."
            }
          }
        }
      ],
      "category": [
        {
          "objectType": "Activity",
          "id": "https://w3id.org/xapi/cmi5/context/categories/cmi5"
        }
      ],
      "extensions": {
        "https://w3id.org/xapi/cmi5/context/extensions/sessionid": "c7b6f0a9-482c-4c03-acc1-548289126963"
      },
      "timestamp": "2016-06-09T15:34:26.887Z"
    }
  }
}
```

# \*Record Scratch\*

SCORM allows passed/failed to be sent repeatedly

Where's the scores?

How do we only send the right one?



# More Impedance Mismatch

Hold the statements back

- ✳ Wait for `cmi.{core.}score.raw`
- ✳ Wait for the first “passed”
- ✳ Wait for LMSFinish / Terminate

Void the statements

- ✳ Requires good LRS support for voiding

No one-size-fits-all solution

# What Else Can We Get?

xAPI SCORM Profile

Interactions

Objectives

Custom xAPI statements

- ✳ So long as they're "cmi5 allowed"

- ✳ Any verb

- ✳ cmi5 context template, no cmi5 category ID

# Terminated

Just like initialized

Don't send anything after this!

Redirect to LMS.LaunchData "returnURL"



# Wrappers

---



# Problems Converting Existing Content

All sorts of things will get in your way with pre-existing content

- ✳ SCORM to cmi5 requires big course changes

- ✳ Weak publishing tool support for cmi5

- ✳ Course publishing JS hard to edit

- ✳ JS object required

  - ✳ SCORM 1.2: API

  - ✳ SCORM 2004: API\_1484\_11

Magic?

---

Intercept those calls!

Replace them with cmi5!

# Building a Wrapper

## Create container HTML page

### ✧ iframe

✧ Don't forget allowfullscreen tag for video!

✧ `<iframe src="index.htm" width="100%" height="100%" style="position:absolute;top:0;bottom:0;left:0;right:0;overflow:hidden;" frameborder="0" scrolling="no" allowfullscreen></iframe>`

### ✧ Popup

✧ Many LMSes already popup once

✧ Confusing experiences

## Virtual SCORM API

---

Provide virtual JS object

SCORM 1.2 (*API*)

or

SCORM 2004 (*API\_1484\_11*)

# SCORM 1.2 Virtual API

---

```
var API = {
  LMSInitialize: function(empty) {

  },
  LMSFinish: function(empty) {

  },
  LMSGetValue: function(key) {

  },
  LMSSetValue: function(key, value) {

  },
  LMSCommit: function(empty) {

  },
  LMSGetLastError: function() {

  },
  LMSGetErrorString: function(errorCode) {

  },
  LMSGetDiagnostic(errorCode) {

  }
};
```

# SCORM 2004 Virtual API

---

```
var API_1484_11 = {
  Initialize: function(empty) {

  },
  Terminate: function(empty) {

  },
  GetValue: function(key) {

  },
  SetValue: function(key, value) {

  },
  Commit: function(empty) {

  },
  GetLastError: function() {

  },
  GetErrorString: function(errorCode) {

  },
  GetDiagnostic(errorCode) {

  }
};
```

## Initialize isn't Special

---

Just put the Statement in there, and done



## To Commit or To Immediate?

---

Commit(“”) requires a fully  
caching adapter.

Don't.

# SetValue Conversion Challenges

Convert calls as they come in

Cache/buffer as needed

- ✱ passed / failed (waiting for score)

- ✱ completed (waiting for passed / failed)

Some keys will need to be stored in different formats

- ✱ Ex: Session / Duration:

- ✱ SCORM 1.2: cmi.core.session\_time: 0000:01:27

- ✱ cmi5: duration (ISO8601): PT0H1M27S

Write Twice, Read Never?

---

Where do we save these?  
Statements **AND** States!

# SetValue Saves

Make Statements where useful

- ✳️ cmi5 required (passed, failed)

- ✳️ Likely reportables (cmi.interactions.n...)

Make States all the time, every SCORM key

- ✳️ More on this in the GetValue section

Misses the read-only keys, but might be good enough

- ✳️ If it's not, you'll have to build a fully conformant adapter; yuck!

# GetValue Gets Expensive

Two approaches to fetching:

- ✱ Pull all potential values up front
- ✱ Fetch as you need them

Two approaches to storing:

- ✱ Fat Storage: All in one key
- ✱ Thin Storage: Each in their own key

SCORM Lives in the Present

---

SCORM is Synchronous!

# GetValue Fetching Ahead of Time

## Steps:

1. Make a storage object
2. Make a list of all the “static” SCORM keys
  1. Regular keys like `cmi.core.score.raw`
  2. Count keys like `cmi.interactions._count`
3. Load these all from the state, store, maybe as separate keys?
4. Iterate through the iterable keys
  1. Ex: `cmi.interactions.n.[id, type, timestamp, weighting, learner_response, result]`
  2. Plus all the nested iterables:
    1. Ex: `cmi.interactions.n.objectives.n.id`

# GetValue Just In Time

Fetch each key when you need it

Synchronous course activity means blocking HTTP requests

✳️ AJAX without the first A!

Could cause stutter in courses that GetValue() often



# GetValue Fat Storage

Store all values in a single JSON object

Store in a single State

Pros:

- ✳ Single State Get gives you all data, less in-course stutter
- ✳ Much less network fetch overhead, especially for courses that reuse the same keys

Cons:

- ✳ Requires a smarter JS API object to manage keys
- ✳ Write network activity could be more expensive if you have a lot of keys; bandwidth based on lots of large requests of mostly unchanging data

# GetValue Thin Storage

Store each value in a separate key









Pros:

- ✳ Can be fetched inline with course, if LRS and network are sufficiently fast
- ✳ For larger sets of keys, much less network write traffic

Cons:

- ✳ More likely to cause in-course stutter

# SetValue and GetValue, Summarized

	Thin Storage	Fat Storage
Low GetValue() Low SetValue()		
Low GetValue() High SetValue()		
High GetValue() Low SetValue()		
High GetValue() High SetValue()		

But can it chop, slice and dice?

---

Still no one size fits all  
solution

# Open Source Wrappers

SCORM to xAPI:

✳ <https://github.com/adlnet/SCORM-to-xAPI-Wrapper>

SCORM to cmi5:

✳ [https://github.com/JCASolutions/scorm\\_to\\_cmi5\\_wrapper](https://github.com/JCASolutions/scorm_to_cmi5_wrapper)

# Wrap Up

SCORM and cmi5 manifests are pretty much the same

Launch puts a little more work on the AU

cmi5 xAPI statements ask for a lot!

Consider using a cmi5 wrapper instead of modifying your course

# Questions?



# Resources

---

cmi5 Specification:

[https://github.com/AICC/CMI-5\\_Spec\\_Current/blob/quartz/cmi5\\_spec.md](https://github.com/AICC/CMI-5_Spec_Current/blob/quartz/cmi5_spec.md)

cmi5 Samples:

[http://aicc.github.io/CMI-5\\_Spec\\_Current/samples/](http://aicc.github.io/CMI-5_Spec_Current/samples/)

xAPI SCORM Profile

<https://github.com/adlnet/xAPI-SCORM-Profile/blob/master/xapi-scorm-profile.md>

Common xAPI verbs:

<http://xapi.vocab.pub/describe/?url=https://w3id.org/xapi/adl>

Contact info:

George Vilches

[gvilches@jcasolutions.com](mailto:gvilches@jcasolutions.com)

321-296-8166 x203