



SCORM 4 Cross Domain (S4XDM)

Full Documentation

Table of Contents

[Introduction](#)

[What is SCORM 4 Cross Domain?](#)

[How SCORM 4 Cross Domain Works](#)

[Technical Details](#)

[Course Setup](#)

[Sample Code](#)

Introduction

The *SCORM 4 Cross Domain (S4XDM) Full Documentation* is designed to provide an in depth explanation of S4XDM. This documentation covers the purpose, use, and technical details of S4XDM. For a quick overview to get started using S4XDM see the *SCORM 4 Cross Domain (S4XDM) Quick Start Guide*.

What is SCORM 4 Cross Domain?

SCORM 4 Cross Domain (S4XDM) is a framework designed to allow SCORM content the ability to communicate across domains without causing cross domain errors, such as 'access denied' with JavaScript. S4XDM has 2 parts; the consumer and the provider. We will go into these parts in more detail in later sections. The consumer is a small SCORM package that is placed on the LMS that you would like to play the content on. The provider is the set of code that is stored on the content server that you would like to play the content from.

There are no specific language or database requirements that need to be met in order to use the SCORM 4 Cross Domain, as this is a client side solution to SCORM cross domain issues. All customizations needed to handle a unique LMS can be set in the configuration file for the stub package, while LMS-agnostic changes can be implemented at various points within the provider. Simple installations often only require setting up the URL that will point to your content server.

SCORM 4 Cross Domain works in conjunction with the Simple SCORM LMS Adapter. The *How SCORM 4 Cross Domain Works* section provides an in depth look at the integration with SSLA.

How SCORM 4 Cross Domain Works

This section covers the SCORM 4 Cross Domain technical details and file workflow. The technical details shown below are the same regardless of whether or not the installation is for SCORM 1.2, SCORM 2004, or both. The file workflow takes the SCORM version into account.

Technical Details

According to the easyXDM website, at the core easyXDM provides a transport stack capable of passing string based messages between two windows, a **consumer** (the main document) and a **provider** (a document included using an iframe). It does this by using one of several available techniques, always selecting the most efficient one for the current browser. For all implementations the transport stack offers bi-directionality, reliability, queueing and sender-verification.

For S4XDM, the **consumer** is a stub package that is uploaded to the LMS and the **provider** is the code on the content server that retrieves the course content from the content server and displays them.

Browser Requirements

According to the easyXDM website, easyXDM provides the following browsers with a latency of less than 15ms:

- IE6 and IE7 – using the FlashTransport (Microsoft Security Bulletin MS11-018)
- IE8+ – using the PostMessageTransport
- Opera 9+ – using the PostMessageTransport (support for both Operas old standard and the HTML5 standard)
- Firefox 1-2 – using the FrameElementTransport
- Firefox 3+ – using the PostMessageTransport
- Safari 4+ – using the PostMessageTransport
- Chrome 2+ – using the PostMessageTransport

SCORM Requirements

Now that we have all possible transport protocols covered we move to the SCORM part. The SCORM part requires 2 pieces, the **stub package** and the **provider**. The stub package is a SCORM content package containing 3 categories of items:

- 1) An `imsmanifest.xml` and accompanying `xsd` files
- 2) A SCORM JavaScript Wrapper and accompanying SCORM API Wrapper files
- 3) The easyXDM library

Manifest file templates are available in S4XDM for use or comparison. The SCORM 2004 and SCORM 1.2 templates can be found respectively at `s4xdm/stub_jca/2004-xsd-manifest/imsmanifest.xml` and `s4xdm/stub_jca/12-xsd-manifest/imsmanifest.xml`.

The fetcher contains the course contents, usually loaded in an iframe, and around that will be wrapped an object called API. This object will be a reference to what the course will believe to be the SCORM API from the LMS, but it will really be an interface to the easyXDM (we will call this API the “Bridge Interface API”) that will talk to the stub package and then the stub package will talk with the actual LMS API Adapter. The runtime and sequence diagrams below define the concept further.

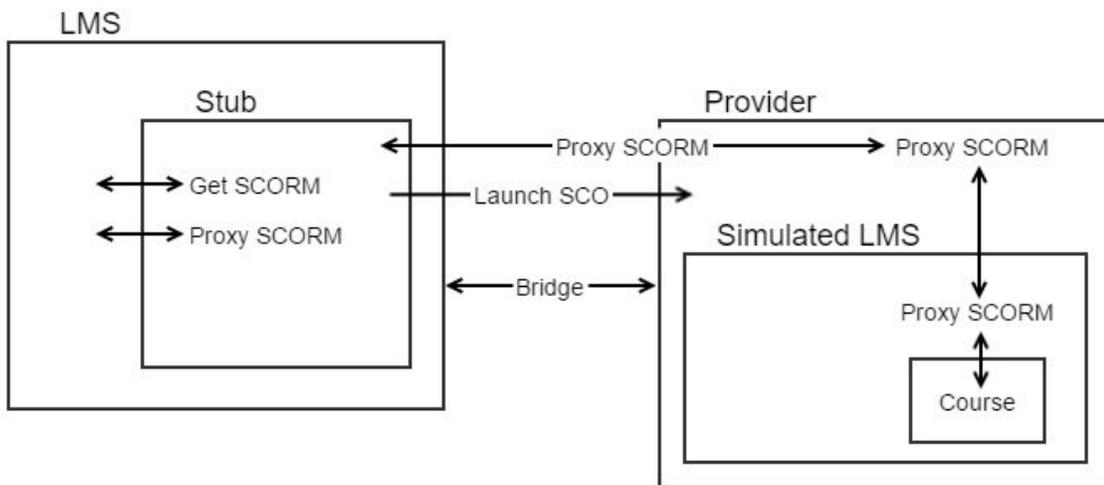


Diagram 1. Runtime

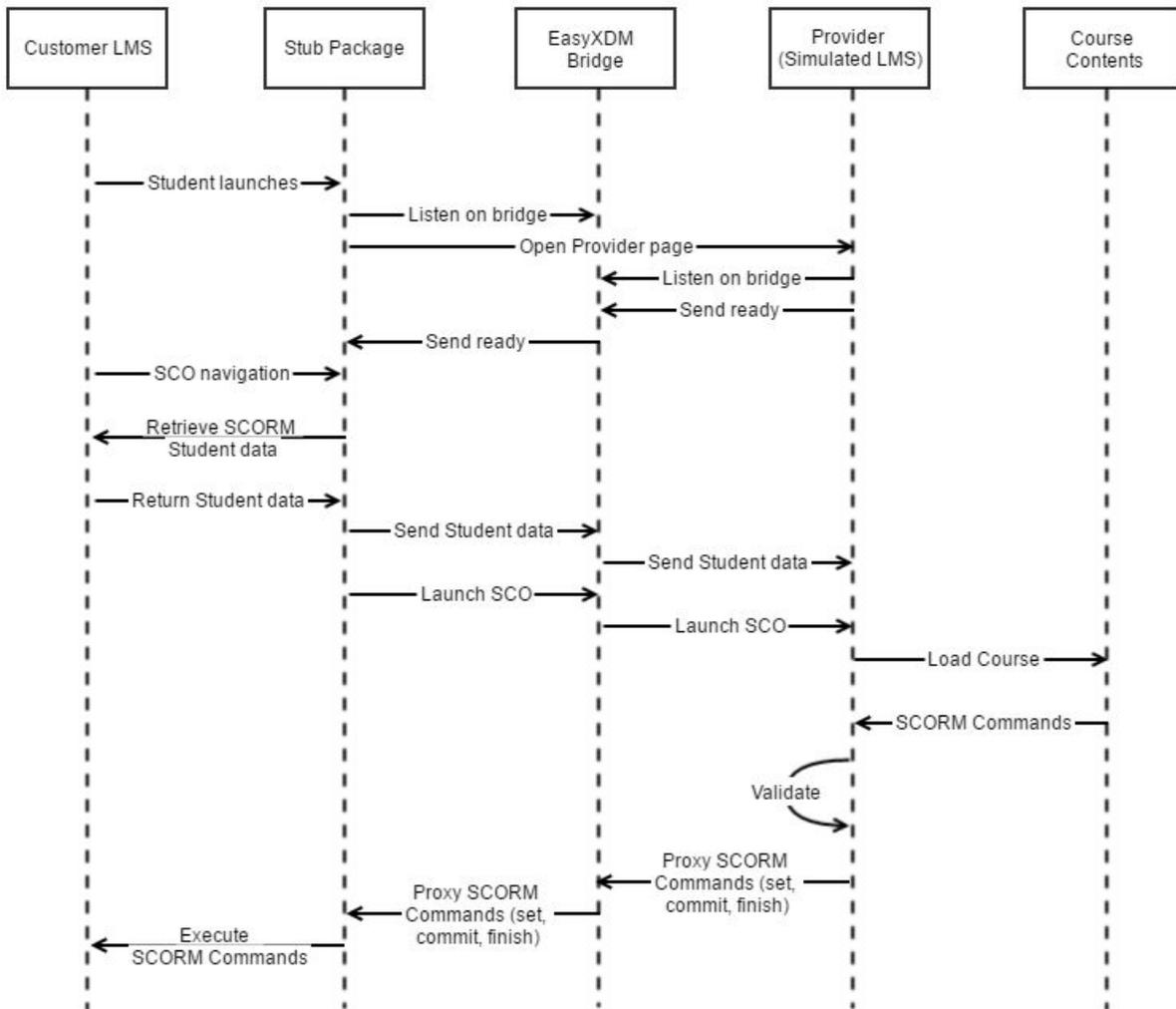


Diagram 2. Sequence

The course is loaded within a SCORM Adapter that exposes a Bridge Interface API. The course thinks the Bridge Interface API is the LMS API Adapter normally exposed by the LMS. The content will find and communicate with the Bridge Interface API. When the Bridge Interface API receives a request to execute a method it communicates with easyXDM and sends the method with the any data to the stub package which also contains an instance of easyXDM. The stub package then sends

the data to the real LMS API Adapter and the stub package sends back a message to the Bridge Interface API that the LMS API Adapter received the message.

If the course needs to get data from the LMS the course makes a SCORM request to get the desired value and requests the data to be sent to it through the Bridge Interface API. That data is forwarded to the stub package. The stub package sends the request to the LMS API Adapter. The LMS API Adapter then returns the data to the stub package, the stub package sends the data to the Bridge Interface API, the Bridge Interface API returns the result to the course and the course reacts to the new data in JavaScript.

Course Setup

These instructions show how to set up the S4XDM fetcher and stub file. Customization techniques are included to allow debugging, posting data and retrieving data.

Extract the S4XDM zip file to a location on your content server where the S4XDM code will have access to your content and the S4XDM is accessible via HTTP. That should be all that you need to do to install the S4XDM on your server.

To set up the stub package to access the location of your S4XDM folder on the server:

1. Open `stub_jca\src\config.js`
2. In the `s4xdmConfig` variable change `providerBaseUrl` so that it points to the fully qualified path for the `_fetcher` directory on your server (e.g. `http://localhost:8000/_fetcher`)
3. You can also change the default content window for your course here if you like by changing these values in the variable `s4xdmConfig`:
 - a. `width=700;`
 - b. `height=700;`

4. Open your imsmanifest.xml file

Change this line in the `imsmanifest.xml` file.

```
<resource identifier="ref001" type="webcontent"
adlcp:scormtype="sco" href="scorm/index.htm?
sco_522765884667=/courses/sco_tester/index.htm">
```

The 'sco_522765884667' is the relative path to the course on your server relative to `providerBaseUrl` you set above. If the relative URL requires parameters (e.g. `/courses/course1/index.htm?keyword=value`), pre-encode those variables with the %-style encoding. Example:

```
?sco_522765884667=/courses/course1/index.htm%3Fkeyword%3Dva
lue
```

Sample Code

```

<?xml version="1.0"?>
<manifest identifier="Manifest_2004-4a" version="1.1"
  xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2
imscp_rootv1p1p2.xsd
  http://www.imsglobal.org/xsd/imsmid_rootv1p2p1
imsmid_rootv1p2p1.xsd
  http://www.adlnet.org/xsd/adlcp_rootv1p2
adlcp_rootv1p2.xsd">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
  </metadata>
  <organizations default="LINEAR">
    <organization identifier="LINEAR">
      <title>Stub Package</title>
      <item identifier="ident001" identifierref="ref001" isvisible="true">
        <title>Stub Package</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="ref001" type="webcontent" adlcp:scormtype="sco"
href="scorm/index.htm?sco_522765884667=/courses/ssl_courses/sco_tester/i
ndex.htm">
      <file href="scorm/index.htm" />
    </resource>
  </resources>
</manifest>

```

Sources

“easyXDM” Retrieved September 27, 2016, from <http://easyxdm.net/wp>