SCORM troubleshooting

Check for problems SCORM on cloud site

To check if the problem is related to Totara or with the SCORM itself you can create an account on https://cloud.scorm.com and upload the SCORM to see if it performs as expected. If it does there may be something wrong with the settings on Totara.

Slow download for first play

Try adding SCORM as an external package type. External package type is disabled by default on new installations; it is not a great method of linking to packages as under certain circumstances your packages will not pass back any grade or tracking information to Totara Learn.

If enabled and selected it provides a text field to paste a URL to a remote imsmanifest.xml (in an unzipped remote SCORM package). This allows SCORM content to be hosted outside your Totara site and the content is played from the external location and is not unpacked/copied into Totara storage. This also means that if you update the SCORM package then the latest content will always be loaded (unless you update your core imsmanifest.xml file as this file is copied into Totara). Be very careful updating your imsmanifest.xml file; if your SCO identifiers change then tracking information for any removed SCO identifiers for any previous user attempts will be cleared from the Totara database.

For the external package type to work your SCORM content must be located under the same URL as your Totara site. If the base URL is different from your Totara URL then built-in web browser security will block cross-domain communication and no grades/tracking information will be passed back to Totara. Adobe Flash and modern browsers (ever since IE 6) prevent cross-domain communication. If you really need your packages to sit outside Totara Learn on a different URL you should look at other options for hosting your content.

To enable the external package type go to *Plugins > Activity modules > SCORM* within the quick-access menu then tick the box for **Enable external package type**.

Debugging

1. Debugging settings are located at *Quick-access menu > Plugins > Activities > SCORM*.
2. Check the checkbox for **Activate API debug.**
3. Set the API mask.

You can use the mask to enable debugging under certain conditions. For example, if you are logged in using the admin user (username admin) you can set the api mask to: *admin.* Users not logged in as admin will not see the debugging log.

What does the debugging log mean?

The SCO commonly sends the following communications through the API:

- **LMSInitialize():** Opens the connection between the SCO and Totara Learn
- **LMSGetValue('valuename'):** Gets a value from Totara Learn
- **LMSSetValue('valuename' , 'value'):** Sends a value to Totara Learn
- **LMSCommit():** Saves values sent to Totara via LMSSetValue() and should be called after every LMSSetValue()
- **LMSFinish():** Saves values sent to Totara and closes the connection between the SCO and Totara Learn

Red lines in the debugging log means there was an error in the communication through the API.

If LMSInitialize() fails, returns an error, then no subsequent values sent to Totara will be saved.

Click through the entire SCORM package. Then access the SCORM report for your attempt, and compare the saved values in the debugging log with Totara's report of the attempt. If the values set in the debug log do not match the values saved to Totara, then there may be a problem with Totara. Otherwise it's likely to be an issue with the SCO or the SCORM activity settings not giving you the functionality you need.

Course goes into review mode when retrying

Once a package has a status of: completed, passed, or failed, then the current attempt has finished. Re-entry to the SCORM package should allow either a new attempt to be generated or the existing previous attempt to be entered into review mode.

If the SCORM content structure page is being displayed and the user enters the SCORM again they will see a checkbox under the enter button that allows them to enter and generate a new attempt. If this checkbox is not ticked it will enter in review mode.

If you are using the student skip content structure page setting then your learners will not see the enter button or the checkbox so to make sure a new attempt is generated you must enable **Force new attempt**.

SCORM results deleted after package update

When uploading a SCORM package over a previous one, if the item identifier in the manifest file is different to the one being overwritten then the tracking data for that SCORM package in Totara is deleted.

Ensure the item identifier is the same for the new SCORM resource.

SCORM and the gradebook

Some SCORM packages report both cmi.core.lesson_status and cmi.core.score.raw. Others report only cmi.core.lesson_status, or only cmi.core.score.raw. The **Grading Method** setting for SCORM objects is meant to account for that.

If you have the **Grading Method** set to **Highest grade**, **Average grade**, or **Sum grade**, and your learning object does not report a score, only, cmi.core.lesson_status, then there will be no numerical score to pass to the gradebook.

If your SCOs do not report cmi.core.score.raw, then the best **Grading Method** setting is **Learning Objects**. This reports either a 1 or a 0 as a score for each learning object. The gradebook value for that SCORM activity is the percent of scos in the package for which learners got a 1.

On the other hand, if your SCOs do not report a lesson_status, then select one of the score-based **Grading Method** options, such as **Highest grade**,**Average grade**, or **Sum grade**, and not **Learning Objects**.

If you do not know what your SCOs are reporting to the LMS, then run them through to completion with debugging on.

Much of the way SCORM objects are graded is controlled inside the SCORM Authoring process before it is packaged for use in an LMS like Totara; make sure all your grading settings are set correctly

My Flash-based content loads, then stalls

This issue is most commonly associated with zlib compression. Classic presentation is that the base HTML file and SWF are loaded into the SCORM player, but the SWF is not able to load any audio or video assets, and therefore stalls. You can watch the loading of SWFs and assets using Chrome's Developer Tools or other.

Check your site's zlib compression settings as an admin by loading up the *Server > phpinfo* page. zlib compression is not a Totara setting but a server setting, so you'll need a server admin to disable it. Be sure to **Purge All Caches**, and clear browser caches, before confirming the change.

Zlib warning when I add/update SCORM objects in my course

Zlib is a php compression setting made in a websites PHP configuration; unfortunately some browsers don't handle this well (especially Internet Explorer 6). Some webhosts enable this setting, but it will likely cause issues for your users when they attempt to view/use the SCORM object. You will need to contact your server administrator to turn this off. The setting to change in php configuration is "zlib.output_compression"

Incorrect file package - missing imsmanifest.xml or AICC structure

This means that Totara cannot find a file called imsmanifest.xml inside the SCORM object. Reasons for this could be:

- imsmanifest.xml needs to be immediately inside the scorm directory, not inside a directory inside of that. So if the zipped SCORM package is package.zip, the unzipped package directory should contain immediately inside of it the imsmanifest.xml. This is a common mistake and normally occurs when a SCORM author creates a package themselves and then selects that folder to compress. This places the content folder inside of another folder. The imsmanifest.xml is there, but it is 2 directories deep. To avoid this problem when zipping SCORM content into a package, go inside of the exported SCORM folder, select all files inside, and compress them while all are selected. The resultant compressed directory has the imsmanifest.xml file in the first directory, immediately available to the Totara SCORM loading process.
- When using Linux based systems the filename imsmanifest.xml must be all in lowercase not IMSmanifest.xml or Imsmanifest.XML
- The SCORM authoring tool Articulate sometimes fails to create the imsmanifest.xml. Try exporting the package again and see if the manifest is generated.
- The SCORM authoring tool Articulate Presenter will publish packages where the imsmanifest.xml file is in the correct place, but there are several lines of white space in the manifest file if you do not fill out the **Reporting and Tracking Options** in Articulate Presenter for Keywords and LMS Description. Totara will give a 'Manifest not found' error when encountering this. To fix this problem select the **Reporting and Tracking Options** in the Articulate Presenter publish dialog and fill in the LMS Description and Keywords.

File not found error

You have this error if the SCORM package is created in Totara, and the SCORM menu loads, but inside of the viewing area for the SCORM content, you get a page with a 404 file not found error, usually showing the current Totara theme.

What this means is that one of the files listed in the imsmanifest.xml is not in the scorm package or not in the correct directory.

Download and unzip the package, open up imsmanifest.xml. At the bottom of the xml file, below any metadata, you'll find a place where organizations and resources for those organizations are designated:

```
<organizations default="TOC1">
         <organization identifier="TOC1">
            <title>SCORM Test</title>
            <item identifier="I_SCO0" identifierref="SCO0">
            <title>Library Quiz</title>
            <adlcp:masteryscore>0</adlcp:masteryscore>
         </item>
         </organization>
      </organizations>
      <resources>
         <resource identifier="SCO0" type="webcontent" adlcp:scormtype="sco" href="scorm.html">
            <file href="scorm.html"/>
            <dependency identifierref="ALLRESOURCES" />
         </resource>
         <resource identifier="ALLRESOURCES" type="webcontent" adlcp:scormtype="asset">
            <file href="scorm.html" />
            <file href="scorm.js" />
            <file href="swfobject.js" />
            <file href="scorm.swf" />
            <file href="scormwrapper.js" />
         </resource>
      </resources>
   </manifest>
```

In this xml, we have a single organization, and in that organization is a single resource, a single SCO. There are 5 files necessary for that resource to work correctly. The scorm.html file is loaded first. It loads 3 external js files and a swf.

What you need to do now is go to the directory containing imsmanifest.xml, and check that all of those listed files are available at the correct path from imsmanfiest.xml, as listed in imsmanifest.xml. If any one of those files is missing (especially the html file or the swf), or if the paths in imsmanifest.xml are incorrect, then it's very likely that the SCORM object won't be able to load at all.

Sometimes the files aren't missing, but are simply misnamed in the manifest, or placed in the wrong directory. You can fix this by moving the files to the correct places, or updating their names so that the imsmanifest and the actual file names match up. In the case of a misnamed file, change the manifest rather than the actual file names, since the the files also reference one another in other places.

Clear an Attempt

To clear attempts by a learner:

1. Go to the SCORM activity and select the link **View reports for x users**.
2. Select the attempt or attempts you want to clear using the checkbox.
3. Select **Delete** in the dropdown box at the bottom of the page.

Handling of Multiple Attempts

SCORM is designed to allow a learner to exit and return at a later date to the same point they left from. This means that each time they enter the SCORM they are using the same single attempt. Some SCORM packages are intelligent about handling re-entry, many are not. What this means is that if the learner re-enters an existing attempt, if the SCORM content does not have internal logic to avoid overwriting cmi.core.lesson_status and cmi.core.score.raw, they can be overwritten by a lower score, confusing the learner.

When a SCORM sets the cmi.core.lesson_status value to 'completed', 'passed' or 'failed' then Totara allows the user to create a new attempt by adding a **Start new attempt** checkbox to the entry page. If cmi.core.lesson_status is set to 'incomplete', 'browsed' or 'notattempted' the learner can only re-enter the existing attempt. If you are using the setting **Student skip content structure page**, this checkbox will never be shown to the user.

Totara provides a range of settings to allow this to be controlled, some of these settings are hidden by default as advanced options.

- **Number of attempts**: This allows the teacher to set how many SCORM attempts the learner may create; this is not how many times a learner can re-enter a SCORM attempt.

- **Attempts grading**: This allows the trainer to set how multiple SCORM attempts (not re-entries) are graded. It is important to note that a 'failed' cmi.core_lesson_status allows a new attempt to be generated but the attempts grading setting **last completed attempt** only includes 'completed' and 'passed' values in it's calculations.

- **Display attempt status**: This displays a users SCORM attempts and how their final grade is calculated on the SCORM entry page and the My Totara page for the learner.

- **Force completed**: This is a setting that can be used to force a SCORM package to report a 'completed' cmi.core.lesson_status if it doesn't currently set the value.

- **Force new attempt**: This hides the **Start new attempt** checkbox and will force a new attempt if the previous attempt has cmi.core.lesson_status value to 'completed', 'passed' or 'failed'. This setting can also be used to make sure a new attempt is generated when the **Student skip content structure page** setting is used.

- **Lock after final attempt**: This prevents access to the SCORM after the total number of attempts have been used. If this is not set the learner can re-enter their last attempt and potentially change/overwrite their score each time depending on how the SCORM package supports multiple re-entries.

Reducing Load Time with Captivate

Modify the percent that must be downloaded before the content starts to play. In Captivate 4, there's a setting in: *Preferences > Project > Start and End > Preload*. Reduce that to 50%.

If you use audio in your Cp file (as background or element attachment), try to put a gap of 0,1 second at the beginning of each element including audio on your slides. If you don't do that, Cp merge all the audio files in one big audio file it need to download before playing the project. This problem have been report many times from the Cp community.

Totara changes cmi.core.lesson_status from 'completed' or 'passed' to 'failed'

Many SCORM authorware suites generate a masteryscore node in the imsmanifest.xml by default. This node is not necessary to the XML file. But when it is there, the SCORM standard designates specific behaviour with regard to the value set there.

Here's what is in the SCORM standard, on page 35 of 155 in SCORM_1.2_ConformanceReq.pdf, numbered page 2-19 (SCORM Version 1.2 Conformance Requirements Version 1.2):

> *If the value for this element is not set to “incomplete” by the SCO, then the LMS shall re-evaluate and change the value based on the following:*
>
> - *If there is no mastery score in the Manifest, and the SCO sets a score (cmi.core.score.raw) and the lesson_status (cmi.core.lesson_status) then the LMS shall not override the SCO determined status.*
> - *If there is a mastery score in the Manifest, the LMS can change the status to either passed or failed depending on the student's score (cmi.core.score.raw) compared to the mastery score.*
> - *If the student is taking the SCO for no-credit, (cmi.core.credit = “no-credit”) there is no change to the lesson_status, with one exception. If the lesson_mode (cmi.core.lesson_mode) is "browse", the lesson_status may change to "browsed" even if the cmi.core.credit is set to no-credit.*

This can result in some functionality you don't intend. You can fix the problem by removing the mastery score node from your imsmanifest.xml file. You will also want to find out what options you have, within your authorware suite, for the writing of that node into the manifest file, and change your authoring process accordingly.

Other items

- Should a problem displaying characters correctly occur, it could be a misconfiguration of the site's server. Make sure that both httpd.conf (when using Apache) and php.ini are set to DefaultCharacterset = utf8 or switch the sending of a default character set off.
- A SCORM package must contains in the root of the zip a file named **imsmanifest.xml** that defines the SCORM course structure.