Diagnosing Elearning Course Completion Issues

Whenever users are reporting issues obtaining the correct status for a course, there are a variety of potential sources of the issue.

LMS	**Course**	**User**
There could be a defect in the LMS itself that is not accurately tracking the status.	The course could be designed in a way that does not conform to the specification for the learning standard or does not return the expected status.	The user or user's environment could be causing issues obtaining the correct status.

It could be one or a combination of any of these sources, which is part of the challenge when attempting to troubleshoot issues.

Basics about elearning courses

All elearning courses are combinations of web-accessible files: HTML, CSS, JavaScript, images, and other types. Essentially, an elearning course is basically a self-contained website that communicates with an LMS using one of the supported standards: SCORM, AICC, xAPI, cmi5, etc. These standards dictate the data elements the course and LMS share with each other, including the appropriate status for a user. Because the course is launched on a user's device, in a web browser, the communication between the course and the LMS happens using JavaScript.

Troubleshooting Reported Issues

When troubleshooting issues reported by a user, there are a number of questions that should be reviewed. This is not an exhaustive list, but provides some items that can be ruled in or out.

Question	What it helps identify
Does the issue affect all users or a subset of users?	The larger the subset of users, the more likely the issue is either the LMS or the Course(s), or both.
Does the issue affect one or more courses?	When more courses are affected, the more likely the issue is either the LMS or User(s) or both.
Does the issue affect all courses for the affected users?	If the affected users have problems with all courses, this points to a User issue or a possible LMS issue.
Are the affected users all using the same browser and version?	Some browsers handle communication slightly differently. Chrome and Edge are based on the Chromium browser base, so they are quite similar. Firefox, Safari, Opera and others have different bases. Asking an affected user to try in a different browser can rule out browser-specific issues.
Are the affected users all accessing the course from the same location(s)?	Networking connections can cause issues with the traffic from end users and the LMS.
Does the company use software to monitor domains?	Services like Cloudflare can have an impact on communication between users and the LMS.
Are the affected users accessing the course via a proxy server?	Proxy servers have settings to cache data, and sometimes affect course communication.

Question	What it helps identify
Do the affected users see any messaging in the course?	Many courses or LMS platforms provide error messages when there are issues within a course. This helps triage possible causes with the LMS or Course(s) or both.
Do the affected users have any browser plugins / extensions installed?	Some browser plugins may affect course communication. To rule these out users can often open the browser in “safe mode” which disables the plugins.
Does the user’s anti-virus software monitor internet communication?	Antivirus software like McAfee can include internet monitoring to ensure protection, which can affect course communication.
Are the affected users attempting to open the course in different browsers or windows?	In some cases, user behavior can be unexpected. We have seen a user open a module in two windows/browsers, using the second window as an “open book” to help answer questions.
When an affected user launches the course, are there any errors in the Developer Console?	Because the communication happens within the browser, any JavaScript errors will show in the Developer Console window.
Has the course been tested in SCORM Cloud?	Testing in SCORM Cloud, as described below, can help verify course communication settings, often ruling out course-related issues.
If the site is integrated with Rustici Engine, has the course been loaded using the Rustici player or the native player?	Testing the course in either player can help isolate an LMS-related or possible course-related issue. **NOTE:** Courses can be migrated from the native player to Rustici Engine, but not the other direction.

Question	What it helps identify
Does the course send `exit=suspend` at the end despite also being completed?	For PFL, the command `exit=suspend` means the attempt is incomplete even if the lesson status or completion status shows completed.
In PFL, is the *Enable Using Last Committed Lesson Status* setting enabled?	By default, the LMS will use the first completed status sent from the course, and will not update after that. Enabling this setting will allow the course to update the status by sending an updated status.
In PFL, are there course settings that might affect the completion set by the course content?	Checking things like mandatory exams, esignature settings, evaluations could block the transcript from showing complete when the content has been completed.

Standard-Specific Troubleshooting

SCORM 1.2

Courses in SCORM 1.2 tend to have fewer course-specific issues. Some things to check with the course.

- **cmi.core.lesson_status** - This will pass the completion status; it should be completed/passed/incomplete/failed/unknown.
- **cmi.core.exit** - This tells the LMS "why" the learner exited the course or SCO. Typically LMS platforms handle exit='suspend' to indicate a user is not finished with an attempt and exit='' to indicate the attempt is finished.

SCORM 2004, 3rd Edition

SCORM 2004, 3rd edition has more items that need to be reviewed. First, within each SCO there are three fields to review.

- **cmi.completion_status** - This is used to indicate completed or incomplete for a SCO.

- **cmi.success_status** - This is used to indicate passed or failed for a SCO.
- **cmi.exit** - This tells the LMS “why” the learner exited the course or SCO. Typically LMS platforms handle exit='suspend' to indicate a user is not finished with an attempt and exit='' to indicate the attempt is finished.

In addition to the course (SCO) communication data, there are some additional items that SCORM 2004 added when using a multi-SCO course. The biggest of which are objectives, rollup rules, and sequencing. These can be found in the course *imsmanifest.xml* file, which is read when the course is loaded into the LMS.

```
<item identifier=”ITEM-1” identifierref=”RES-1”>
 <title>Module 1</title>
 <imsss:sequencing>
  <imsss:deliveryControls completionSetByContent="true"
    objectiveSetByContent="true"/>
  <imsss:objectives>
    <imsss:primaryObjective objectiveID="">
      <imsss:mapInfo targetObjectiveID="" readSatisfiedStatus="true"
      writeSatisfiedStatus="false"/>
      <imsss:minNormalizedMeasure>0.8</imsss:minNormalizedMeasure>
    </imsss:primaryObjective>
  </imsss:objectives>
  <imsss:rollupRules>
    <imsss:rollupRule minimumCount=”1”>
    <imsss:rollupConditions>
      <imsss:rollupCondition condition=”satisfied”/>
    </imsss:rollupConditions>
     <imsss:rollupAction action=”completed”/>
    </imsss:rollupRule>
    </imsss:rollupRules>
  </imsss:sequencing>
</item>
```

Many of these are optional, and it is possible to have a course with sequencing and without rollup rules, but not vice versa. Some key things to review:

- **CompletionSetByContent=true** - This will be in the manifest under deliveryControls to indicate the course will tell the LMS when the user is complete. If this is missing, the course will mark complete as soon as it is launched.
- **minNormalizedMeasure** - This will be the passing score for the course in decimal format (0.8 means 80% passing score).

NOTE: This section will expand as more SCORM 2004 manifest settings are reviewed.

AICC

Like SCORM 1.2, the AICC model (cmi4 and before) is fairly straightforward. The only issue to review would be in tne

- **lesson_status** - This is nearly identical to the cmi.core.lesson_status, reflects the learner's completion status for the lesson; it should be completed/passed/incomplete/failed/unknown.

xAPI

PFL supports xAPI courses in both the native player and Rustici Engine.

NOTE: This section will be detailed when more xAPI course issues are reviewed for possible inclusion.

cmi5

Currently, PFL only supports cmi5 courses using Rustici Engine as the course player. Although cmi5 is a variation of xAPI, the native player has not been updated to include this variation.

NOTE: This section will be detailed when more cmi5 course issues are reviewed for possible inclusion.

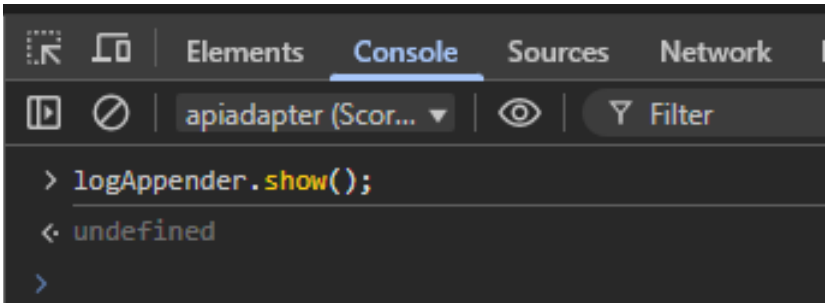
PFL Course Communications

Within PFL, customers have the ability to use either the native player or use Rustici Engine as the player if the site is integrated with Rustici Engine. Having this log output can help diagnose issues. Look for the completion or lesson status in the console.

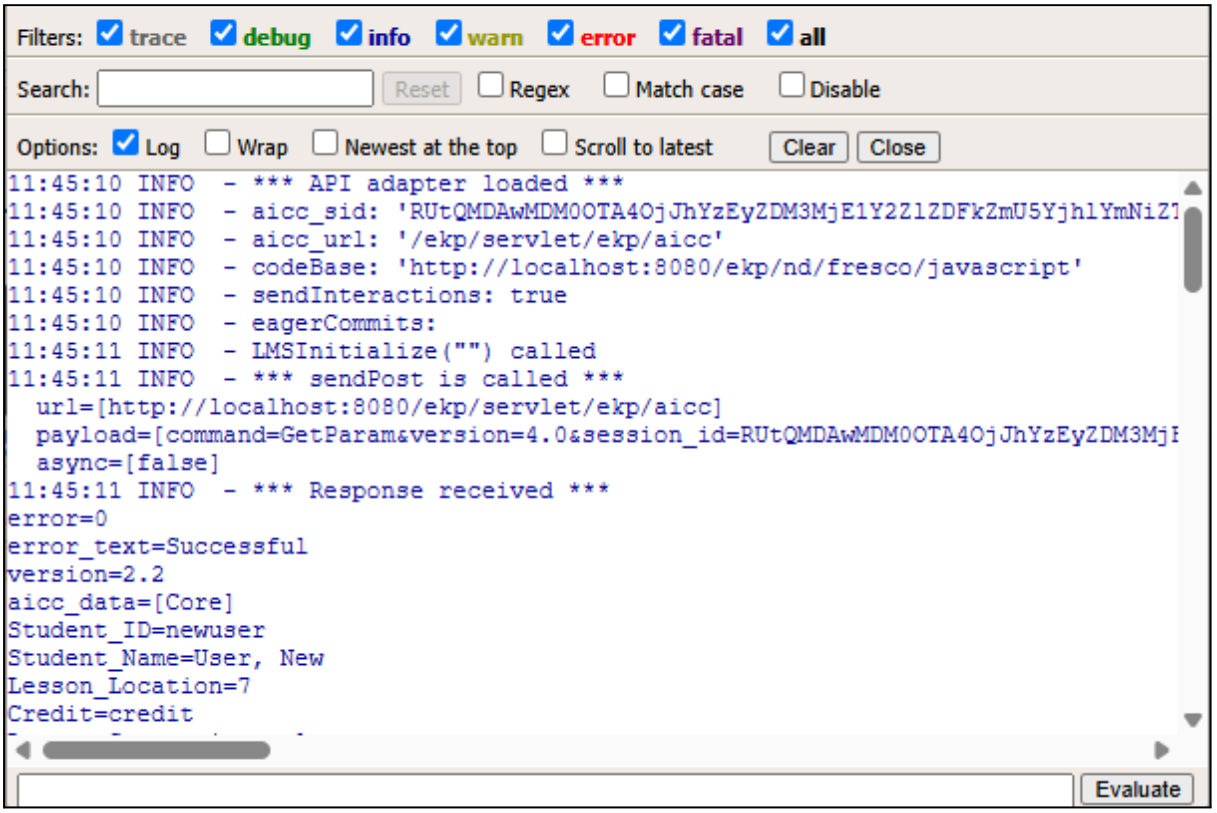
PeopleFluent's native adapter

To open the adapter for a course that uses the native player:

- Open the browser's Developer Tools (e.g. Ctrl+Shift+I on Google Chrome).
- In the Developer Tools, switch to the Console tab.
- In the selector at the top left, change from *top* to *apiadapter*.
- In the console, enter the following text: logAppender.show();



This will pop up the log window, which will close with the course window.



Rustici Engine Adapter

For customers on sites integrated with Rustici Engine and where a course is launching using Rustici Engine as the launch interface (player), it also has an adapter to see communication. To open the adapter for a course that uses the Rustici Engine player:

- Open the browser's Developer Tools (e.g. Ctrl+Shift+I on Google Chrome).
- In the Developer Tools, switch to the Console tab.
- In the console, enter the following text: Debug.ShowAllAvailableData();
- NOTE: Depending on how the course is set up to launch, you may need

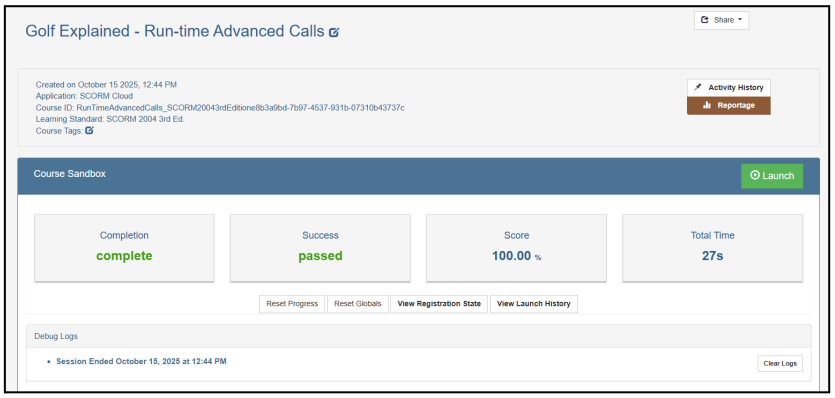
to change the selector to ensure you are targeting the Engine frame [not pictured].

This will pop up the log window which will close with the course window.

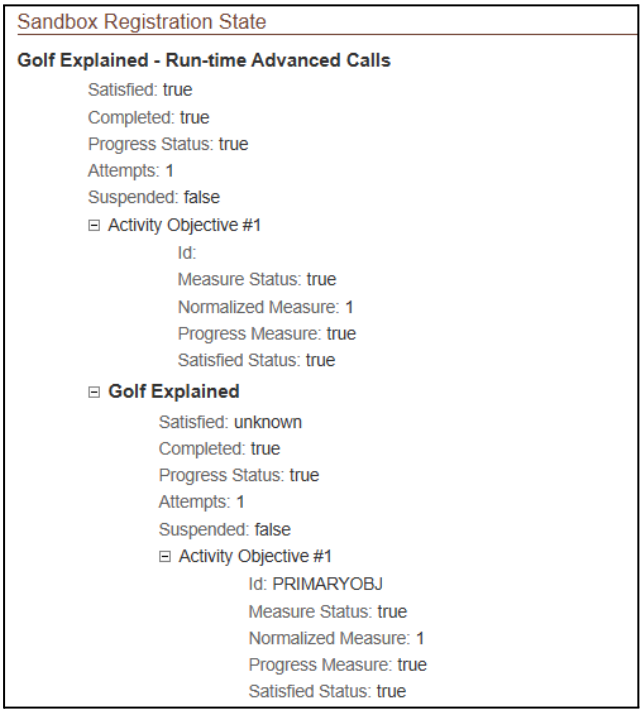
SCORM Cloud

Regardless of whether the issue might be LMS, Course, or User, one initial step to validate communication is to load the course into SCORM Cloud [https://cloud.scorm.com] and test it.

NOTE: SCORM Cloud allows free accounts as long as the total size of the course library is less than 3 courses and 5GB total size; PeopleFluent can help obtain a license for a larger library if need be.



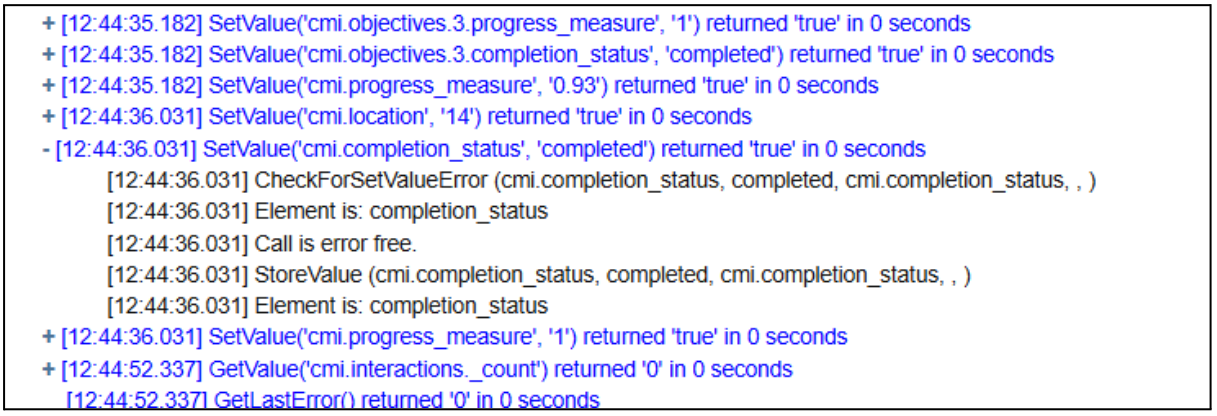
View Registration State



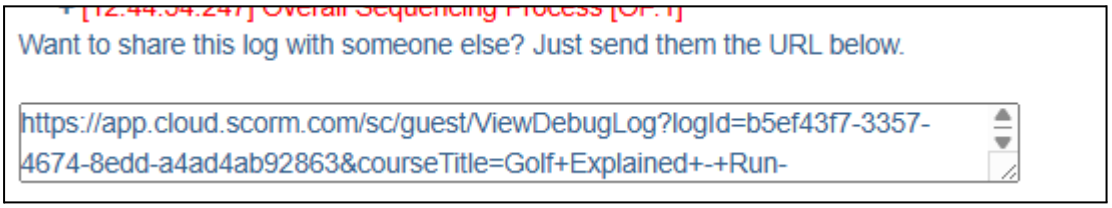
The Registration State is a current view of the details for the current registration. Viewing the information can be helpful, especially for multi-SCO courses with sequencing. In the view check for **Satisfied** and **Completed** at the top and within each SCO.

Debug Logs

The log shows the SCORM communication, so it may not be as helpful for AICC or xAPI/cmi5 courses. However the status fields appropriate to the standard ,such as **cmi.completion_status**, can be seen.



One benefit of SCORM Cloud Debug logs: they can be shared with others who do not have the course. At the bottom of the log is a URL.



This can be helpful when submitting support tickets.

PFL Database

Within PFL, there are a few tables where course communication settings are stored, as well as the user's transcript information. For PeopleFluent SaaS customers, our Hosting team can run SQL to get information. However, for team members testing locally or for PFL sites hosted elsewhere here are the tables and fields to review.

AICC_core - This table stores the data coming from the elearning course, but is also used for other course types.

- session_id
- userid
- courseid
- lesson_status
- completionStatus
- successStatus
- lastAttemptDate
- score1
- highestscore

AICC_core_detail - This can help align the course launches with the transcript history dates.

- session_id
- Launch_date

AICC_Core_Trans_Attempts - This links the AICC Core data to a specific transcript record.

- session_id
- transcript_id

AICC_student_data - This shows each attempt on the course, and the status for each lesson.

- session_id
- status
- score

transcript

- learningid
- userid
- transcriptid
- studentstatus
- final_score
- last_updated
- contentSessionID
- attempts_count

transcript_history

- transcript_id
- user_id
- learning_id
- original_status
- resultant_status
- last_updated

It is possible to construct SQL to join these tables together, to get the correct data to match. For example, matching the transcript id and session ids to limit to specific users and courses. However, the potential SQL will depend on the desired data.

PFL Auditing & Debug Mode

If completions happen more frequently, it may be beneficial to enable auditing and use the R409 report to review activity. Specifically, the suggestion would be enable auditing and debug mode:

- **Audit Option** (Configured in ekp.properties on the server) - Course Launch
- **Audit Level** (Can be configured in the UI) - High
- **Debugging** (Configured in the UI) - On

NOTE: The Audit and Debug settings above can have an impact on performance, so it may only be temporarily enabled while working with an affected learner. Also, these changes may not provide a final insight, but the data may be useful in suggesting alternative investigation paths.

Accessing the R409 report and the EKP.LOG for the time period when testing can be helpful to review potential issues.

Other Resources

Here are some other resources that can be helpful.

SCORM.com

Rustici Software supports a site with a wide range of articles about the SCORM standard and SCORM courses. [https://scorm.com]

AI Tools

Using artificial intelligence tools like Google Gemini, Chat GPT, or Claude can also help provide steps to diagnose potential problems related to courses and elearning standards. They will be less helpful, or not at all helpful, for issues related to the LMS.

When submitting a prompt, be sure to include any relevant information about the standard and the information users are seeing. AI should not be the first place for diagnosing issues, but it can help if you've identified issues through other means.

Rustici Support

While this is not an option for Partners or Customers, PeopleFluent's support team can reach out to Rustici Software for assistance with course-related issues – even if the course is not using Rustici Engine as the player. Rustici cannot diagnose issues within the LMS itself, which will require assistance from PeopleFluent team members.