

Project Number: NTH - AAQL

Adding Interoperability to ASSISTments:  
Learning Tools Interoperability (LTI) Support and  
Partial Credit

A Major Qualifying Project Report

Submitted on December 18, 2014

To the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE



In partial fulfillment of the requirements for the

Degree of Bachelor of Science

In Computer Science

By:

---

Nicholas Massa

Advised by Neil Heffernan

This report goes over the process of creating support for the LTI API. LTI is a set of standards to allow educational websites to communicate with each other. LTI uses Tools to authenticate and send user data from one site to another and with extensions can support the transfer of grades. This will allow researchers to use ASSISTment's assignments and features from their educational site of choice. This will also allow ASSISTments teachers to assign content to users from other educational websites. In addition support for partial credit has been provided to reports.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction to LTI .....                              | 5  |
| 2. Security (OAuth).....                                  | 5  |
| 3. LTI v1.0 and v1.1 .....                                | 5  |
| 4. Project Design .....                                   | 6  |
| 4.1 Providing ASSISTments as a Tool Provider.....         | 6  |
| 4.2 Providing ASSISTments as a Tool Consumer.....         | 7  |
| 4.3 Partial Credit for ASSISTments.....                   | 14 |
| 5. Project Implementation.....                            | 18 |
| 5.1 Overview .....  | 18 |
| 5.2 Implementation of ASSISTments as a Tool Provider..... | 18 |
| 5.3 Implementation of ASSISTments as a Tool Consumer..... | 18 |
| 5.4 Implementation of Partial Credit.....                 | 22 |
| 6. Future Work .....                                      | 24 |
| 7. Conclusion.....  | 24 |
| 8. Appendix.....  | 25 |
| 9. References .....                                       | 55 |

## List of Figures

|  |    |
|--|----|
| Figure 1 – New Beta/Research LTI Option .....            | 8  |
| Figure 2 – LTI Creation Button .....                     | 8  |
| Figure 3 – LTI Creation Page.....                        | 9  |
| Figure 4 – LTI in Problem Sets Folder.....               | 9  |
| Figure 5 – Teacher Page Assigning LTI .....              | 10 |
| Figure 6 - Redirecting Page when Launching LTI .....     | 10 |
| Figure 7 - Pop-up for an Example LTI Producer .....      | 11 |
| Figure 8 – Student Page with LTI.....                    | 11 |
| Figure 9 - Grade Returned .....                          | 12 |
| Figure 10 - Class Report with LTI.....                   | 12 |
| Figure 11 - Student Report with LTI .....                | 12 |
| Figure 12 - Proficiency Report with LTI.....             | 13 |
| Figure 13 - Item Report with LTI .....                   | 13 |
| Figure 14 - Tracker for Partial Credit.....              | 14 |
| Figure 15 – New Partial Credit Preference and Link ..... | 15 |
| Figure 16 – Partial Credit Settings Page.....            | 16 |
| Figure 17 - Teacher Item Report .....                    | 16 |
| Figure 18 - Individual Student Report.....               | 17 |
| Figure 19 - Student's Item Report.....                   | 17 |

## Acknowledgements

Thank you to Professor Neil T. Heffernan, Cristina Heffernan, David Magid, and Andrew Burnett for their time, support, feedback, and input while I developed LTI and Partial Credit support for ASSISTments. It is with their professional help that I was able to create and program these features.

Thank you to Siyuan Zhao for introducing me to LTI, ASSISTments, and web development. Thank you to Doug Selent for helping me manage code that interacts with the database. And thanks to the ASSISTments team that provided support and help throughout.

# Learning Tools Interoperability (LTI) Support and Partial Credit

By Nicholas Massa

Advised by Neil Heffernan

**Abstract.** LTI can provide support for ASSISTments to interact with any educational website, without having to specifically implement an API for each site. Researchers from sites such as EdX can assign content from ASSISTments using this standard, without any need for the two websites to structure an API for the other. Partial Credit will allow students who did not answer a question correctly on their first try to be given a percentage of points.

## 1. Introduction to LTI

Learning Tools Interoperability is a standard specified by the IMS Global Learning Consortium. In LTI the learning applications are described as Tools. Tool Consumers are the Learning Management Systems (LMS) that send user data over to a learning application. Tool Providers are the learning application that receives the data. LTI can be implemented to provide ASSISTment's features to other educational applications, and to allow interoperability from ASSISTments to other educational websites.

## 2. Security (OAuth)

The LTI specification works to provide secure yet seamless connections from the Consumers to the Providers. The main authentication tool for these connections is OAuth. OAuth is an open standard to authorization. It is designed to be used over HTTP connections and provides access tokens. These tokens are created and then sent to the connecting website, where they may be accepted or rejected. Tokens can be rejected based on timestamps that differ too greatly from one site to the other, an invalid secret key being passed in, or a random Nonce number that the connecting website has already used. OAuth is commonly used by third party sites, who would like to allow users to use their Google or Facebook accounts safely on the third party site. The third party site may use the information to create an account that the user can log into simply with their Google or Facebook account. This is similar to LTI, where a teacher's or student's information is securely passed to a third party site.

## 3. LTI v1.0 and v1.1

LTI version 1.0 is the most basic implementation of LTI. It is the most commonly supported version of LTI and supports a secure link from a Tool Consumer. LTI requests are made from HTTP POSTs, and contain a series of parameters. A variety of parameters are passed in, including information about the Tool Consumer, a user's name email and their roles on the Tool Consumer's site, and values for

OAuth authentication. The IMS Global Consortium provides examples of the request and an optional response: [1](#)

In version 1.1 the ability to pass back, read, update, and delete grades was added. The IMS Global Consortium provides a very detailed and final draft of LTI 1.1: [2](#)

LTI v1.2 and LTI v2.0 have recently been proposed by the Consortium. As of yet few LTI Providers support the new features in these specifications. For this project two packages of software have been used to easily provide OAuth authentication and LTI support up to LTI v1.1. When these packages are updated to the new specifications, ASSISTments shall be up to date as well. There may be more parameters to send from ASSISTments as a Tool Consumer, however.

## 4. Project Design

### 4.1 Providing ASSISTments as a Tool Provider

The first goal of this project was to implement ASSISTments as a Tool Provider. This allows users and researchers to use the educational assignments and reports within ASSISTments from their own websites or applications, such as EdX.

Currently ASSISTments has built an Edmodo API which allows Edmodo users to be linked into ASSISTments classes and assignments. This is done through a Java servlet service, which accepts incoming requests and processes the information. The Edmodo service was written by the Masters student Hien D. Duong, and more information from the project can be found here: [3](#)

In essence, supporting LTI is an extenuation of Hien's project to use feature-oriented software engineering to integrate ASSISTments with other LMSs. As opposed to directly supporting a specific site however, ASSISTments is able to provide content to any LTI v1.1 compliant Tool Consumer in a manner very similar to Hien's.

While Hien's code has been updated and modified to support LTI, the SPV Java LTI Tool Provider package has been used, under the GNU Lesser General Public License. More information about this tool is available here: [4](#)

Using SPV Software Products, the details for implementing and performing OAuth authentication is handled for us. Along with this classes are provided to more easily access the data from LTI requests. A Tool Consumer object is built based on the incoming request, and then a Tool Provider object is executed to verify and authenticate the Tool Consumer.

To learn how to create experiments in ASSISTments using LTI view the following link: [5](#)

The page above contains examples of ASSISTments being used as a Tool Provider. Please view the Moodle and EdX examples to gain a better understanding of the implementation.

## 4.2 Providing ASSISTments as a Tool Consumer

The second goal of this project was to implement ASSISTments as a Tool Consumer. This allows users and researchers to use external assignments and websites from within ASSISTments.

Implementing ASSISTments as a Tool Consumer is quite different from implementing the system as a Tool Provider. Critical information and changes from within ASSISTments was needed to provide this feature. Plus, SPV Software Products does not currently support a Java implementation to run a server as a Tool Consumer.

Instead the Ruby library `ims-lti` was utilized. This library is Copyright © 2012 Instructure, but full permission to use the software is granted as long as the `ims-lti` license is provided in copies of ASSISTments. Along with this, the OAuth library was also installed, in order to build and authenticate requests and responses. More details on the library can be found: [67](#)

As LTI links are in essence external assignments and links for users, it is best to implement these links as an assignment. These assignments do not contain any problems, but contain the required LTI parameters, such as the title, launch url, key, secret, and any custom parameters. To create these assignments modifications to the builder were necessary. After building the link, the LTI assignment can then be assigned to students. Although not all Tool Providers will return a grade, ASSISTments can listen in to see if a student has had a grade returned after doing an assignment. If so, it will store it with the student's assignment log. Teachers will be able to see the grade in reports, while students can see the grade next to the assignment link on their page. Certain features are made unavailable with LTI assignments, such as previewing a link or test driving a link before it is assigned.

Assistment Teacher Student Builder Admin Develop Account Nick (nbmassa@wpi.edu) Logout  
 Preferences Need help?

---

Developer Settings

On Off Features  
 Show Unencoded IDs

Save Preferences Restore Defaults

---

Teacher Settings

On Off Features  
 Do Not Show Teacher in Student Reports  
 Assigning to sections or individuals  
 Send me a message when my students complete past due assignments  
 Show "Problems" link (Builder tab)  
 The default standards for skill domains

Hints will not be available until the student has made  attempt(s) to answer the question.  
 After the first hint has been selected, delay the availability of each additional hint by  seconds.

The default time for assignment release dates is:    
 The default time for assignment due dates is:

[\[-\] Beta/Research Features](#)

- Do not show Youtube videos as the web page support during in-school hours (6:00am-3:00pm).
- Show attempts count option when creating your own Problem Sets
- Show "Assemble Problem Set Using Existing Problems" link (Builder tab)
- Show "LTI" Link (Builder Tab)
- Show "Create Skill Builder Problem Set" link (Builder tab)
- Show "Skills" link (Builder tab)
- Can Add Sections to Problem Sets

Save Preferences Restore Defaults

Developed At: Worcester Polytechnic Institute Created in Collaboration with: Carnegie Mellon

Figure 1 – New Beta/Research LTI Option

To create an LTI link, one must enable Show "LTI" Link in the Beta/Research Features.

Assistment Teacher Student **Builder** Admin Develop Account Nick (nbmassa@wpi.edu) Logout  
 Problem Sets Search Skills Preferences Need help?

Problem Set ID:  [Edit](#) - [New Copy](#) - [Details](#) - [View Problems](#) - [Test Drive](#)

[+ Create Your Own Problem Set](#) [+ Create Your Own LTI Link](#)

Problem Sets I've Built Close all

Problem Sets I've Built Create Sub-folder

PSAP3D7 - Neil Test Problems [Assign](#) - [Edit](#) - [Move](#) - [Bookmark](#) - [View Problems](#) - [Test Drive](#)

Figure 2 – LTI Creation Button

After enabling the setting, a 'Create Your Own LTI Link' appears



**Assistment** Teacher Student **Builder** Admin Develop Account Nick (nbmassa@wpi.edu) Logout

Problem Sets Search Skills Preferences Need help?

LTI is used to connect assignments to ASSISTments.  
 ASSISTments will share the launched username and e-mail with the tool.  
 Please make sure to enter a valid URL, along with any extra necessary information.  
 Custom parameters and their values are split by '=', each parameter must be on a newline. Ex:  
 param1=value  
 param2=value2

Invalid parameters will be ignored.

Title:  The Title can be anything

Launch URL:  The Launch URL right now is my own local Tool Provider.

Consumer Key:  The key between my Consumer and Provider is 'testing'

Shared Secret:  The secret between my Consumer and Provider is 'secret'

Custom Parameters:  The custom parameter appid is set to the value of 15.  
 Multiple parameters can be entered, seperated by a newline.

Figure 3 – LTI Creation Page

This is the creation, copying, and editing page for LTI links.

**Assistment** Teacher Student **Builder** Admin Develop Account Nick (nbmassa@wpi.edu) Logout

Problem Sets Search Skills Preferences Need help?

Problem Set ID:  [Edit](#) - [New Copy](#) - [Details](#) - [View Problems](#) - [Test Drive](#)

**Problem Sets I've Built** [Close all](#)

PSAQ8FS - My Assignment [Assign](#) - [Edit](#) - [Move](#) - [Bookmark](#)

PSAP3D7 - Neil Test Problems [Assign](#) - [Edit](#) - [Move](#) - [Bookmark](#) - [View Problems](#) - [Test Drive](#)

Figure 4 – LTI in Problem Sets Folder

The problem set in in purple is an LTI link. It can be assigned, edited, moved, and bookmarked similar to other problem sets. However, attempting to View Problems or Test Drive the set will result in a message that there are no problems and return the user back.

Figure 5 – Teacher Page Assigning LTI

Assigning an LTI link is very similar to any other problem set.

Figure 6 - Redirecting Page when Launching LTI

Upon assigning it the Teacher will be redirected to the LTI Provider. A pop-up of assistments.org/teacher will come up, so the Teacher does not have to click back to return to ASSISTments.

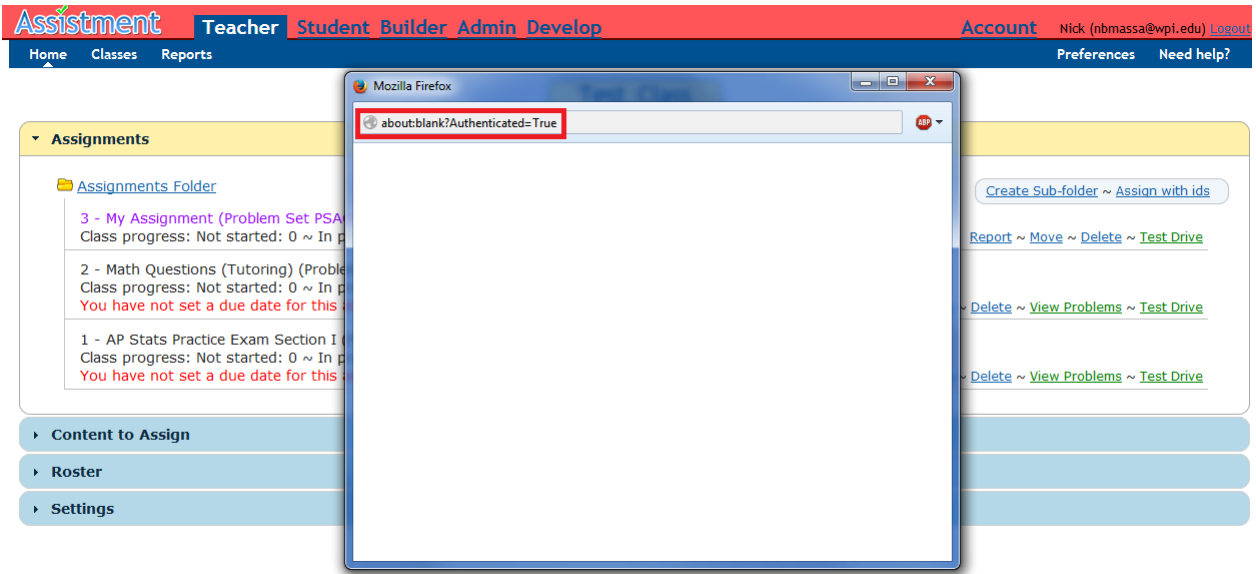


Figure 7 - Pop-up for an Example LTI Producer

If the Teacher 'Test Drives' the assignment, a pop-up will open connecting them to the LTI Producer. This LTI Producer is an example that shows that OAuth authentication worked correctly.

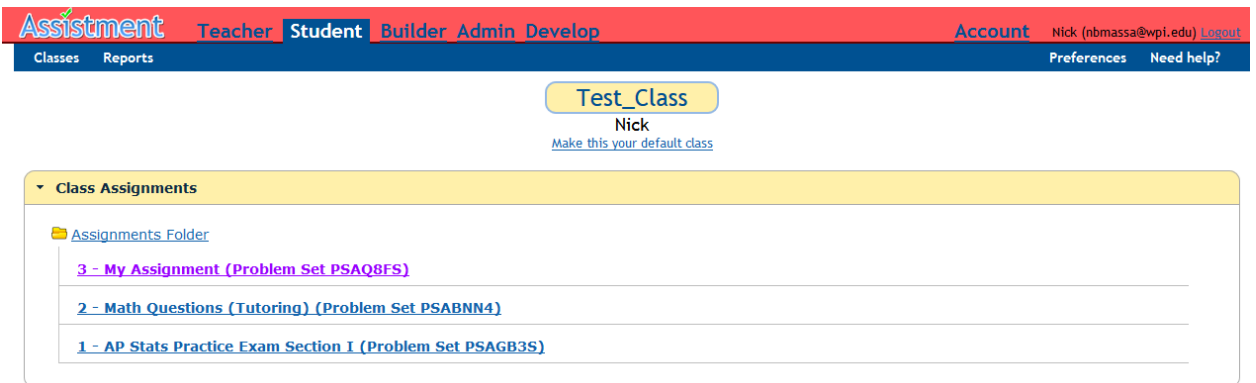


Figure 8 – Student Page with LTI

A student's LTI assignment can be clicked on just like any other assignment.

Test\_Class  
 Nick  
[Make this your default class](#)

Class Assignments

Assignments Folder

- 3 - My Assignment (Problem Set PSAQ8FS) Grade: 0.60
- 2 - Math Questions (Tutoring) (Problem Set PSABNN4)
- 1 - AP Stats Practice Exam Section I (Problem Set PSAGB3S)

Figure 9 - Grade Returned

After working, if the LTI Provider returns a grade, ASSISTments will receive and store the grade in a problem log. The student's grade is made available on the right of the assignment link.

Test\_Class

Assignment Report

Class: Test\_Class Download Report to: [PDF](#) ~ [Spreadsheet](#)  
[\[+\] Customize This Report](#)

Showing the 20 most recently assigned problem sets

| Student<br><a href="#">[Anonymize]</a> | <a href="#">PSAQ8FS</a><br>-----<br>3 - My Assignment | <a href="#">PSABNN4</a><br>-----<br>2 - Math Questions (Tutoring) | <a href="#">PSAGB3S</a><br>-----<br>1 - AP Stats Practice Exam Section I |
|--|---|---|--|
| Average                                | 60%   | 0%  | 0%   |
| Complete                               | 0%  | 0%  | 0%   |
| Massa, Nicholas B                      | In Progress   |   |  |

Figure 10 - Class Report with LTI

Here, PSAQ8FS is an LTI assignment. The grade from all students is formatted and calculated into an average, allowing Teachers to view an average of their LTI assignment.

Individual Student Report

Class: Test\_Class Download Report to: [PDF](#) ~ [Spreadsheet](#)  
[\[+\] Customize This Report](#)

Student: Nicholas Massa

| Assignment   | Completion Status | Grade                               | Last Worked On             | Type of Assignment  |
|--|-------------------|-------------------------------------|----------------------------|---------------------|
| <a href="#">3 - My Assignment</a>                    | In progress       | / - 0.60<br><a href="#">Details</a> | <a href="#">11-11-2014</a> | External Assignment |
| <a href="#">2 - Math Questions (Tutoring)</a>        |                   | 0/0 - 0%<br><a href="#">Details</a> |                            | Regular Problem Set |
| <a href="#">1 - AP Stats Practice Exam Section I</a> |                   | 0/0 - 0%<br><a href="#">Details</a> |                            | Regular Problem Set |

Figure 11 - Student Report with LTI

An individual student's LTI grade will be displayed without any formatting.

Assistent Teacher Student Builder Admin Develop Account Nick (nbmassa@wpi.edu) Logout

Home Classes Reports Preferences Need help?

Test\_Class

Can't generate proficiency report from an external Assignment. Please see Individual Student Reports or Class Level Assignment Reports. [\[+\] Legend](#)

Proficiency Report

Section: Entire Class

Assignment: 3 - My Assignment (Problem Set PSAQ8FS)

[Details](#)  
[View Problems](#)  
[Print this report](#)

| Student            | Original questions |         |         | All questions |         |         |             |
|--------------------|--------------------|---------|---------|---------------|---------|---------|-------------|
|                    | Done               | Correct | Average | Done          | Correct | Average | Total hints |
| Massa, Nicholas B. |                    |         |         |               |         |         |             |
| Class average      | 0                  |         | 0%      |               |         |         | 0           |

Figure 12 - Proficiency Report with LTI

Trying to view a proficiency report for an LTI assignment will issue a warning message that LTI assignments can't be used in a proficiency report.

Assistent Teacher Student Builder Admin Develop Account Nick (nbmassa@wpi.edu) Logout

Home Classes Reports Preferences Need help?

Test\_Class

External assignment item reports are not available. Please see Individual Student Reports or Class Level Assignment Reports. [\[+\] Legend](#)

Item Report

Class: Test\_Class

Section: Entire Class

Automatically refresh report every 2 minutes. [Refresh Now](#) Download Report to: [Spreadsheet](#)

Hide Correct Answers Row

Assignment: 3 - My Assignment (Problem Set PSAQ8FS)

| Student/Problem                       | Average     | Total Hints | Time Spent |
|---------------------------------------|-------------|-------------|------------|
| [Anonymize]                           | Data driven |             |            |
| Problem Average <a href="#">Graph</a> |             |             |            |
| Common Wrong Answers                  |             |             |            |
| Correct Answer(s)                     |             |             |            |
| Massa, Nicholas B.                    |             | 0           | N/A        |

No skill report available by your settings.

[Graph Student Average](#)

[\[+\] Advanced](#)

Figure 13 - Item Report with LTI

Similar to proficiency reports, trying to view the item report for an LTI assignment will issue a warning message that LTI assignments can't be used in an item report.

### 4.3 Partial Credit for ASSISTments

The third goal of this project was to provide partial credit to ASSISTments. This allows students to feel less discouraged after getting a problem wrong, gives students points for continuing to get it correct without too many attempts or hints, and provides a different analysis for teachers in reports.

Implementing Partial Credit into ASSISTments involved quite a bit of modification to the Report Controllers and Report Helpers, as well as a few changes to some Ruby webpages.

#### Description

Default teacher setting for Partial Credit on Feature is "3", when it is off it should be set to "1"

Give a teacher preference for partial credit in the item report. Always calculate the partial credit for every problem even if the feature is not turned on. Hide the column for partial credit if the feature is turned off. When the feature is turned on show the partial credit column but include a checkbox at the top of the item report that allows the users to toggle between partial credit score and non-partial credit score. When the user checks this box, hide the columns for partial credit and show the columns for non-partial credit.

For non-multiple choice questions the algorithm for the partial credit will be  $1 - (x-1)(1/y)$  where x is the number of attempts and/or hints used and y is the maximum number of attempts/hints set by the teacher.

If it is a multiple choice question, this will be the algorithm:

Number of options affects the size of the penalty. Use the formula  $1/(n-1)$  where n is the number of options and the result is the amount subtracted from full credit for each attempt.

Green Check - Correct on first attempt

Green x - Correct on subsequent attempt as long as the number of attempts is 1 less than the maximum number of possibilities and the student has not exceeded the maximum attempts/hints set by the teacher

Red x - Exceeded max attempts and/or reached the last possible choice

Red x with yellow - Bottom out hint

#### Attachments

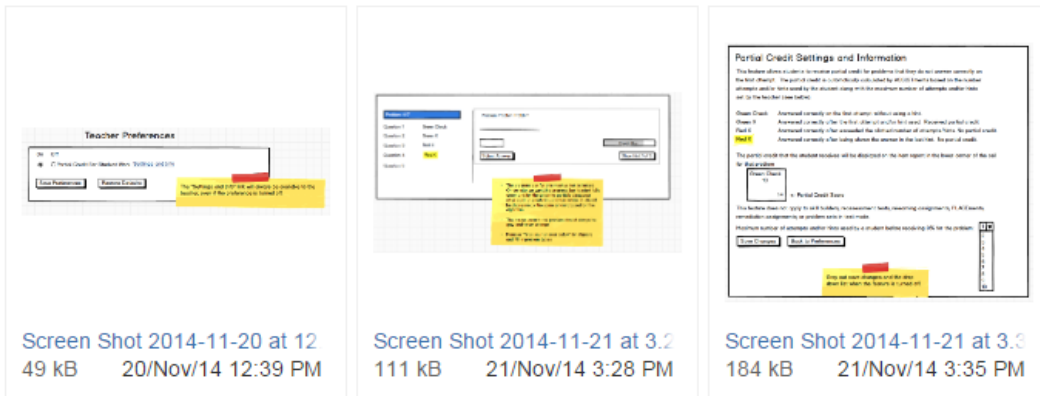


Figure 14 - Tracker for Partial Credit

Assistment [Teacher](#) [Student](#) [Builder](#) [Admin](#) [Develop](#) [Account](#) Nick (nbmassa@wpi.edu) [Logout](#)

[Inbox](#) [Outbox](#) [Send](#) [Preferences](#) [Need help?](#)

### Developer Settings

On Off Features  
  Show Unencoded IDs

[Save Preferences](#) [Restore Defaults](#)

### Teacher Settings

On Off Features  
  Do Not Show Teacher in Student Reports  
 **Partial Credit For Student Work [Credit Settings and Info](#)**  
  Assigning to sections or individuals  
  Send me a message when my students complete past due assignments  
  Show "Problems" link (Builder tab)

Hints will not be available until the student has made  attempt(s) to answer the question.

After the first hint has been selected, delay the availability of each additional hint by  seconds.

The default time for assignment release dates is:

The default time for assignment due dates is:

The default standards for skill domains :

Math

English Language Arts

[\[+\] Beta/Research Features](#)

[Save Preferences](#) [Restore Defaults](#)

Figure 15 – New Partial Credit Preference and Link

There are two new preferences for Partial Credit. One is a credit setting for it to be turned on or off. The other is a number for the allowed maximum attempts or hints. A link to credit settings and information is provided and can always be clicked.

Assistment [Teacher](#) [Student](#) [Builder](#) [Admin](#) [Develop](#) [Account](#) Nick (nbmassa@wpi.edu) [Logout](#)

[Home](#) [Inbox](#) [Outbox](#) [Send](#) [Preferences](#) [Need help?](#)

## Partial Credit Settings and Information

This feature allows students to receive partial credit for problems that they do not answer correctly on the first attempt. The partial credit is automatically calculated by ASSISTments based on the number attempts and/or hints used by the student along with the maximum number of attempts and/or hints set by the teacher (see below)

- ✔ Answered correctly after the first attempt without using a hint.
- ✘ Answered correctly after the first attempt and/or hint used. Received partial credit.
- ✘ Answered correctly after exceeded the allotted number of attempts/hints. No partial credit.
- ✘ Answered correctly after being shown the answer in the last hint. No partial credit.

The partial credit that the student receives will be displayed on the item report in the lower corner of the cell for that problem.

This feature does not apply to skill builders, reassessment tests, relearning assignments, PLACements, remediation assignments, or problem sets in test mode.

Maximum number of attempts and/or hints used by a student before receiving 0% for the problem:

[Save Changes](#) [Back to Preferences](#)

Figure 16 – Partial Credit Settings Page

This page explains in detail how the Partial Credit system works, and should be read by everyone. If Save Changes is clicked the credit setting will be turned on and the maximum number of attempts or hints will be updated.

Test\_Class

**Item Report**  
 Class: Test\_Class  
 Section: Entire Class ▾

Automatically refresh report every 2 minutes. Refresh Now
Download Report to: [Spreadsheet](#)

Hide Correct Answers Row

Assignment: 4 - Multiple Skills 3rd Grade (1) - 14 Questions (Problem Set PSAH9T)

| Student/Problem<br>---<br>[Anonymize]    | Average<br>---<br>Data driven | PRAB76Z<br>---<br>Data driven | PRAB76P<br>---<br>Data driven | PRAB76R<br>---<br>Data driven | PRAB76U<br>---<br>Data driven | PRAB76V<br>---<br>Data driven | PRAB794<br>---<br>Data driven | PRAB76J<br>---<br>Data driven | PRAB796<br>---<br>Data driven | PRAB76Y<br>---<br>Data driven |
|--|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| Problem Average<br><a href="#">Graph</a> | 21%                           | 0%                            |                               | 39%                           |                               | 0%                            |                               | 66%                           |                               | 0%                            |
| Common Wrong<br>Answers                  |                               |                               |                               |                               |                               |                               |                               |                               |                               |                               |
| Correct Answer(s)                        |                               | 27 ÷ 3 = _                    | 2 and 3/4                     | 150                           | 4                             | June 2                        | add 105                       | 6 rows of 3<br>pictures       | 6879                          | Spot , Wags                   |
| <a href="#">Massa, Nicholas B</a>        | 21%<br>                       | ✗ 27 x 3 = _<br>0%            |                               | ✗ 123<br>39%                  |                               | ✗ Hint<br>requested<br>0%     |                               | ✗ Hint<br>requested<br>66%    |                               | ✗ Hint<br>requested<br>0%     |

No skill report available by your settings.

[Graph Student Average](#)

[\[+\] Advanced](#)

Figure 17 - Teacher Item Report

This example of a Teacher’s item report displays the new features of the partial credit system. The percentage earned for a student and their problem appears below each icon. A new green X shows that the student used either multiple attempts or a hint, and therefore received partial credit.



## Individual Student Report

Class: Test\_Class

Download Report to: [PDF](#) - [Spreadsheet](#)

[\[+\] Customize This Report](#)

Student: Nicholas Massa

| Assignment   | Completion Status | Grade                                | Last Worked On | Type of Assignment  |
|--|-------------------|--------------------------------------|----------------|---------------------|
| <a href="#">4 - Multiple Skills 3rd Grade (1) - 14 Questions</a> |                   | 0/5 - 21%<br><a href="#">Details</a> | 12-13-2014     | Regular Problem Set |
| <a href="#">3 - Adding and Subtracting Expressions</a>           |                   | 0/0 - 0%<br><a href="#">Details</a>  |                | Regular Problem Set |
| <a href="#">2 - Math Questions (Tutoring)</a>                    |                   | 0/0 - 0%<br><a href="#">Details</a>  |                | Regular Problem Set |
| <a href="#">1 - AP Stats Practice Exam Section I</a>             |                   | 0/0 - 0%<br><a href="#">Details</a>  |                | Regular Problem Set |

Figure 18 - Individual Student Report

The individual student report has been updated to show the percent of grade earned by the student. The correct-problems/total-problems remain unchanged.

Test\_Class

Nick

[Make this your default class](#)

### 4 - Multiple Skills 3rd Grade (1) - 14 Questions

| Problem                 | My Answer             | My Classmates' Average | Hint Usage |
|-------------------------|-----------------------|------------------------|------------|
| <a href="#">PRAB76J</a> | Hint requested<br>66% | 66%                    | 1          |
| <a href="#">PRAB76R</a> | 123<br>39%            | 39%                    | 0          |
| <a href="#">PRAB76Y</a> | Hint requested<br>0%  | 0%                     | 2          |
| <a href="#">PRAB76V</a> | Hint requested<br>0%  | 0%                     | 2          |
| <a href="#">PRAB76Z</a> | 27 x 3 = _<br>0%      | 0%                     | 2          |
| PRAB796                 |                       | N/A                    | 0          |
| My average: 21%         |                       |                        |            |
| Class average: 21%      |                       |                        |            |

Figure 19 - Student's Item Report

The student report has been updated to be more similar to the Teacher's item report. Partial credit is calculated and displayed below each icon for each answer.

## 5. Project Implementation

### 5.1 Overview

This section is intended for developers and programmers, especially members of the ASSISTments team that will have to develop and support these features. The Tool Provider was implemented in Javascript and relies on Hien's system for external accounts. The Tool Consumer and partial credit system was written in Ruby on Rails and Rhtml.

Currently ASSISTment's Tool Provider is up and running as an existing Java service. Integration as a Tool Consumer and implementing the partial credit system has yet to be made, as there will be some merging of this code with the code in production. Many changes to existing files are only minimal changes, checking if an assignment is an LTI assignment and processing an action slightly differently. For testing I used PostgreSQL on the testing database as16/as14, and I ran a local ASSISTments server.

### 5.2 Implementation of ASSISTments as a Tool Provider

As mentioned before, the implementation of ASSISTments as a Tool Provider is based heavily on Hien's code. As of the date of this report, Siyuan Zhao is maintaining the LTI Servlet.

The Servlet uses the SPV library heavily for authenticating incoming LTI requests. After the connection is authenticated the LTI user's role is used to determine their ASSISTments role. A utility has been written that takes the LTI information and either creates or retrieves an ASSISTments account, returning an access token. After this the user's class is either created if they are the teacher launching the link for the first time, or it is retrieved, using the LTI variables 'context\_title' and 'context\_id', which are the same for all teachers and students using a LTI link in a specific section. The custom parameter 'custom\_appid' is used to determine what problem set is being requested, and the problem set is either assigned if the teacher is launching the link for the first time, or it is retrieved. After this the user is directed to ASSISTments where they will land on the external teacher or tutor page.

### 5.3 Implementation of ASSISTments as a Tool Consumer

All changes made take place within the app itself, as well as a few tweaks to a few Ruby tools.

The changes required the following libraries:

[IMS LTI for Ruby:](#)

[Oauth for Ruby:](#)

**Changes to the Ruby packages in as\_core were made to support Ruby 1.8.6:**

**Oauth-0.4.7 -> Net\_http.rb**

Requests used an unsupported string function to check that the header was 'x-www-form-urlencoded'

Modified to only check the request has content to be valid for authentication.

### **Actionpack-1.13.3 -> Cgi\_process.rb**

Added a method for returning the 'body' and 'url' of a cgi request, so requests can be passed in for authentication by the Oauth tool.

The 'url' is built from a combination of environment variables, and the body is the 'raw\_post\_data' environment variable.

### **Changes to the Ruby source in ast:**

#### **App\Controlllers\Build\Sequence\_controller.rb**

New method: createlti

This method is used to create, edit, or copy lti links. It checks to see if it is part of an existing link and if it was called upon to make a copy, and builds the link accordingly. Unlike a regular sequence, an LTI sequence has no problems, and the following data is stored:

sequence[:parameters]["lti\_title"] = Passed in Title

sequence[:parameters]["lti\_launchurl"] = Passed in URL

sequence[:parameters]["lti\_key"] = Passed in Key

sequence[:parameters]["lti\_secret"] = Passed in Secret

sequence[:parameters]["lti\_params"] = Passed in Custom parameters

sequence[:parameters]["is\_link"] = "true"

Modified method: show

Show was modified to check if the sequence passed in is an LTI link. If it is, it redirects to the LTI page accordingly with the existing parameter. This handles edits to LTI links.

Modified method: duplicate

Duplicate was modified to check if the sequence passed in is an LTI link. If it is, it redirects to the LTI page accordingly with the existing and copy parameters. This handles copies to LTI links.

#### **App\Controlllers\api2\api\_helper\_controller.rb**

New method: grade\_passback

This method takes a request for a grade, authenticates it, and then either stores a sent grade, clears the grade in the database, or returns the stored grade in the database.

#### **App\Controlllers\Teacher\folder\_controller.rb**

Modified methods: create\_class\_assignment, assign\_to\_sections\_or\_individuals, assign\_folder\_item\_to\_class

These methods were modified to check if an LTI assignment was being assigned. If so, then the tool launches the LTI link forcing the teacher to visit the external page, and it does not try to render any more content.

Modified method: assign\_folder\_item

When creating an LTI link from the previous methods, the value @lti\_link is created. This value is checked in this method to ensure that it does not try to render a template after a redirect.

#### **App\Controlllers\Teacher\report\_controller.rb**

Modified method: class\_level\_summary

Checks to see if an LTI assignment was passed for a proficiency report, and flashes a warning that this report can't be made from an LTI assignment.

Modified method: item\_level

Checks to see if an LTI assignment was passed for an item report, and flashes a warning that this report can't be made from an LTI assignment.

Modified method: assignment\_level\_progress

Checks to see if an assignment being calculated is an LTI link, and if so uses the stored data as a grade. This allows teachers to see a class average of LTI assignments.

Modified method: individual\_student\_level\_progress

Checks to see if an assignment being calculated is an LTI link, and if so it uses the stored grade as a percentage. This allows teachers to see a specific student's LTI grade.

### **App\Controllers\Tutor\Student\_class\_controller.rb**

New method: tool\_launch

This method is used to launch an LTI link, redirecting the user from ASSISTments to the website. It requires 'assignmentID' as a parameter. It then builds all the required LTI parameters before executing the redirect.

### **App\Helpers\application\_helper.rb**

New method: render\_lti\_link

This method takes in a class assignment, a display name, and a color and returns a link that will launch an LTI site in a new window.

New method: render\_lti\_grade

This method takes in a class assignment and, if a grade for the current user exists, it returns that grade.

New method: render\_lti\_link\_student

Similar to render\_lti\_link, but adds a parameter studentlaunch=true for when the user clicks the LTI link. This is so that even if a user has a role that is above a learner, they can still test their link as a student from the student page.

### **App\Models\sequence.rb**

New method: lti\_link?

Similar to the mastery\_learning? method, lti\_link? checks to see if the sequence has the 'is\_link' parameter. If it does, it returns true, otherwise false.

### **App\Models\class\_assignment.rb**

New method: lti\_link?

Similar to the mastery\_learning? method, this lti\_link? method checks first if the sequence is blank; if it has a sequence then it calls the sequence's lti\_link? method and returns that.

### **App\Views\Build\Sequence\index.rhtml**

Added a new table element to the div "new\_sequence\_link"

This element is the button to 'Create Your Own LTI Link'

It checks to see if the `show_builder_lti_link` setting is created and true first, if not the button is not displayed.

#### **App\Views\Build\Sequence\createlti.rhtml**

This is a new page is used for creating an LTI link. It contains fields for a title, launch url, consumer key, shared secret, and custom parameters.

It is also used to edit or copy an LTI link. If the controller sees that parameters have been sent requesting an edit or copy of an existing link, it will set `@variables` that this page can use to display the current link's data.

#### **App\Views\Folder\\_assign\_form.rhtml**

When assigning an assignment, a check is added if the item sequence is an LTI link. If it is, then a pop-up is open to `assistments.org/teacher`. This is required as the main ruby page will redirect the user to the LTI link.

#### **App\Views\Teacher\Folder\\_class\_assignment.rhtml**

Modified to check if an assignment being displayed in a folder is an LTI link. If it is not an LTI link, it renders a 'View Problems' and 'Test Drive' link with the assignment as normal. Otherwise it creates a 'Test Drive' link that connects the teacher to the site.

#### **App\Views\Teacher\Folder\\_class\_assignment\_dates.rhtml**

Modified to hide the warning for a due date not being set if the assignment is an LTI link.

#### **App\Views\Teacher\student\_class\folder\\_sequence\_actions.rhtml**

Modified to check that an assignment is an LTI link when placing it in a folder for assigning. If it is not an LTI link, it creates a 'View Problems' and 'Test Drive' link as normal.

#### **App\Views\Tutor\Folder\\_lti\_class\_assignment.rhtml**

A new file for rendering an LTI link in the student's assignment list. If there is a grade stored, it will display it on the right of the assignment.

#### **App\Views\Tutor\Folder\\_class\_assignment.rhtml**

Modified to check if a class assignment is an LTI link, and if it is renders the `_lti_class_assignment.rhtml` page instead of the `_reg_class_assignment.rhtml` page.

#### **App\Views\Tutor\student\_class\tool\_launch.rhtml**

The page that is loaded before an LTI link is opened. It can be set so that the user clicks a "Launch the tool" button before being redirected. Currently it is set to automatically launch the link.

#### **Db\Migrate\330\_create\_show\_lti\_link\_in\_teacher\_setting.rb**

This is the migrate code to create the `show_builder_lti_link` setting, which determines whether or not to display the LTI creation button in the builder.

The following link is an LTI example for Ruby Sinatra that proved very helpful for development: [8](#)

The `tool_launch.rhtml` page was based heavily upon a `tool_launch.erb` page from their example, and the `tool_consumer.rb` file helped to develop the `grade_passback` and `tool_launch` methods in the `ASSISTments` controllers.

## 5.4 Implementation of Partial Credit

All changes made take place within the app itself. Most of the changes focus on the Report Helpers and Report Controllers. Sections that have been changed to support partial credit for, in general, have been commented with “#Partial Credit”.

### **App\Controllers\Teacher\report\_controller.rb**

New method: `partial_credit_from_log` log

This method takes a problem log and calculates the partial credit based on the student and problem id. It checks if the Hash `@student_partial_credit` is created and instantiates it if not. It checks to see if the partial credit was already calculated. If not the log is processed for a partial credit score, and the score is saved into the map at `[student.to_s + "/" + problem_id.to_s]`. Otherwise it pulls the value from the hash. The score is decided based on the lower value between the multiple choice calculation and the attempt/hint total calculation.

Modified method: `assignment_level_progress`

The `@average` hash now uses the `partial_credit_from_log` function to calculate correctness for an assignment/user.

Modified method: `individual_student_level_progress`

The `@student_info` hash now uses the `partial_credit_from_log` function to calculate the percent correct for an assignment.

### **App\helpers\teacher\report\_helper.rb**

New method: `partial_credit` student, problem

This method takes a student and a problem\_id and calculates the partial credit. The behavior is the same as from the `partial_credit_from_log` function described above, except that it retrieves the log instead of being passed in one.

Modified method: `get_problem_average` problem

This method now iterates through all the students and calculates the problem average based on the partial credit.

Modified method: `get_student_average` student

This method now iterates through all the problems and calculates the student's average based on the partial credit.

Modified method: `get_class_average`

This method now iterates through both all the students and all the problems, calculating the average for the class.

Modified method: `get_student_answer_text`

Underneath the answer icon, the percent score the student received will now be displayed.

### **App\helpers\tutor\report\_helper.rb**

The changes to the tutor's report helper are incredibly similar to the teacher's report helper, with differences being mostly in variable names. Like the teacher's helper, the tutor's helper's `partial_credit`, `get_problem_average`, `get_student_average`, and `get_class_average` have been modified.

Modified method: `get_student_correctness`

Based on the attempts and hints used, the method has been changed to display the partial credit icons, including the green X, if the teacher has turned on partial credit.

#### **App\Controllers\Teacher\credit\_controller.rb**

This is the controller for the Partial Credit Settings and Information page. It notifies the `rhtml` page of the number of `credit_attempts` the user has set, default 3, and it saves the changes to any of the settings.

#### **App\Views\Teacher\Credit\credit\_settings.rhtml**

This is the Partial Credit Settings and Information page. It provides all the information on partial credit, and it provides 1-10 max attempts for the Teacher. When the Teacher saves the page, their `credit_setting` is set to true, and their `credit_attempts` is set to the selected value from a drop-down box.

#### **App\Views\Preferences\\_teacher.rhtml**

The loop iterating through each of the Teacher's preferences has been modified. If the setting is 'credit\_attempts', it is hidden. If the setting is 'credit\_settings' a link to the Partial Credit Settings and Information page is added.

#### **App\Models\student\_class.rb**

New method: `credit_setting`

Returns the value of the credit setting for the teacher (true or false)

#### **App\Views\Teacher\\_submenu.rhtml**

Modified to check if there is a `@home_link_only` parameter. If so it does not display the classes and reports menu. Used for the Teacher's credit page.

#### **Db\Migrate\329\_create\_credit\_settings.rb**

This is the migrate code to create the `credit_setting` and `credit_attempts` settings, which determines whether or not partial credit is enabled, and the maximum number of hint/attempts for the partial credit.

There are also two new images created:

**Public\images\report\incorrect\_green\_small.png** and  
**Public\images\report\incorrect\_yelshadow\_small.png**

## 6. Future Work

There is much potential for future work in these systems:

LTI and OAuth versions will be updated though the years to come and ASSISTments will need to update the implementation to stay up to date. Along with this, the IMS global consortium provides IMS certificates for systems that implement LTI and other protocols correctly. To become LTI certified, ASSISTments will need to become an IMS member or participant, and then pass a series of tests. More information is available at: [9](#)

The tutor should be updated to display partial credit icons as the student works on the assignment. An experienced GWT developer should perform this task.

Partial credit can be calculated in a variety of ways, and may see even more advanced calculating techniques, especially if new problem types are created.

As reports are changed, modified, and added, partial credit will need to be used in new and different ways.

## 7. Conclusion

LTI can be used to help students and teachers alike. LTI allows users from one application to use content from another without needing to create and keep track of a new account. This protocol supports grade returns, so after a user works on content their grade can be passed back to the first application. ASSISTments has been turned into both a Tool Provider and Tool Consumer, as it can take in users from LTI websites or it can send off its own users as LTI users. This allows teachers and students who already use an existing system such as Moodle to take advantage of the analytic features in ASSISTments, and allows ASSISTments users to use content beyond the problems that ASSISTments provides.

Partial credit provides more than an alternate grade and new pictures for teachers and students. Partial credit will provide more motivation for a student to solve a problem correctly after making a mistake, instead of simply using all their hints or making attempts with wild guesses. Teachers will be able to use ASSISTments to more fairly evaluate and judge the performance of their class. Partial credit provides new insight for learning research and statistics.



## 8. Appendix

This section contains only the new and modified code - it may not contain all the code in the file. Ellipses (...) are used to designate unmodified code.

### Java Servlet Code:

#### Connect.java

```
@WebServlet({ "/Connect", "/connect" })
public class Connect extends HttpServlet {

    private static final long serialVersionUID = 1L;
    public static final String partnerRef = "Hien-Ref";

    public Connect() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        DataConnector dc = new None();

        ToolConsumer toolConsumer = new ToolConsumer("testing.edu", dc, true);
        toolConsumer.setName("Testing");
        toolConsumer.setSecret("secret");
        toolConsumer.save();

        Callback doLaunch = new DoLaunch();
        ToolProvider toolProvider = new ToolProvider(request, response, doLaunch, dc);

        toolProvider.execute();
    }
}

class DoLaunch implements Callback {

    @Override
    public boolean execute(ToolProvider toolProvider) {
        String userType = "";
        String onBehalf = "";
        String userRef = "";
        User ltiUser = toolProvider.getUser();
        ResourceLink resourceLink = toolProvider.getResourceLink();

        if(true) return true;

        //Determine user role from LTI
        if(ltiUser.isAdmin() || ltiUser.isStaff()) {
            userType = Constants.TEACHER_ROLE;
        } else if ( ltiUser.isLearner() ) {
        // } else {
            userType = Constants.STUDENT_ROLE;
        }
    }
}
```

```

//If user exists, get user ref and access token from the table, otherwise, create a new account, user ref and
access token
ArrayList<String> userRef_and_token = Utility.transferUser(toolProvider, ltiUser.getId(), ltiUser.getId());

userRef = userRef_and_token.get(0);
onBehalf = userRef_and_token.get(1);
//onBehalf = "4486de28-7758-4eb2-95c3-1d7dbc70a663";
toolProvider.setRedirectUrl(goInsideASSISTments(userType, onBehalf));

//Assign Edmodo school to user
String school_ref = Utility.EDMODO_SCHOOL_REF;
SchoolController.assignUserToSchool(userRef, school_ref, Connect.partnerRef, onBehalf);

String contextTitle = toolProvider.getRequest().getParameter("context_title");
String contextID = toolProvider.getRequest().getParameter("context_id");
if(contextTitle == null || contextTitle.equals("")) {
    if(contextID != null) {
        contextTitle = contextID;
    } else if(resourceLink.getTitle() != null && resourceLink.getTitle().equals("")) {
        contextTitle = resourceLink.getTitle();
    } else {
        contextTitle = "Default Class";
    }
}
contextID = resourceLink.getId();
String appId = toolProvider.getRequest().getParameter("custom_appid");
//convert it into problem set number id
String problemSetID = Utility.decodeProblemSetString(appId);
//If teacher, try to create class. Otherwise enroll in the class.
if(userType.equals(Constants.TEACHER_ROLE)) {

    String classRef = Utility.createClass(contextTitle, contextID, Utility.PARTNER_REFERENCE,
onBehalf);

    //Create assignment if given an appId
    if(problemSetID !=null && problemSetID.length()>0){
        Utility.createAssignment(problemSetID, classRef, onBehalf);
        toolProvider.setRedirectUrl(Utility.getUserPageURL(userType, Connect.partnerRef,
onBehalf, problemSetID));
    }
} else if(userType.equals(Constants.STUDENT_ROLE)) {
    String classRef = Utility.createClass(contextTitle, contextID, Utility.PARTNER_REFERENCE,
onBehalf);

    HttpSession session = toolProvider.getRequest().getSession();
    session.setAttribute("user", ltiUser);
    session.setAttribute("resourceLink", resourceLink);
//    Utility.createAssignment(problemSetID, classRef, onBehalf);
    StudentClassController.enrollStudent(classRef, userRef, Connect.partnerRef, onBehalf);
    //Goto assignment if given an appId
    if(problemSetID !=null && problemSetID.length()>0){
        toolProvider.setRedirectUrl(Utility.getUserPageURL(userType, Connect.partnerRef,
onBehalf, problemSetID));
    }
}

//Grade Return
// String grade = assisTmentsUserController.getGradeAverage(userRef);
// Outcome outcome = new Outcome();

```

```

//      outcome.setValue(grade);
//      outcome.setType(ResourceLink.EXT_TYPE_DECIMAL);
//      resoucelink.doOutcomesService(ResourceLink.EXT_WRITE,outcome,toolProvider.getUser());

        return true;
    }

    public static String goInsideASSISTments(String userType, String onBehalf) {
        String addressToGo = "";
        String onFailure = ApplicationSettings.ASSISTments_Login_Failed;
//      String onExit = ApplicationSettings.Connector_URL+"gradeRequest";
//      String onExit = "https://test1.assistments.org/teacher";
        String from = "LTI";
        if (userType.equals(ApplicationSettings.edmodo_teacher_constant_str)) {
            String onSuccess = "https://www.assistments.org/teacher";

            addressToGo =
String.format("%1$s?partner=%2$s&access=%3$s&on_success=%4$s&on_failure=%5$s&from=%6$s",
ApplicationSettings.ASSITments_Login_WebAPI,ApplicationSettings.partner_reference,onBehalf, onSuccess, onFailure,
from);
        }else if (userType.equals(ApplicationSettings.edmodo_student_constant_str)) {
            String onSuccess = "https://www.assistments.org/tutor";
            try {
                onSuccess = URLEncoder.encode(onSuccess, "UTF-8");
            } catch (Exception e) {
                e.printStackTrace();
            }
            addressToGo =
String.format("%1$s?partner=%2$s&access=%3$s&on_success=%4$s&on_failure=%5$s&from=%6$s",
ApplicationSettings.ASSITments_Login_WebAPI,ApplicationSettings.partner_reference,onBehalf, onSuccess, onFailure,
from);
        }
        return addressToGo;
    }
}

```

#### Utility.java

```

public class Utility {

    public static final String PARTNER_REFERENCE = "Hien-Ref";
    public static final String PARTNER_ID = "3";
    public static final int CLASS_SECTION = 2;
    public static final String COURSE_NUMBER = "1";
    public static final String SECTION_NUMBER = "1";
    public static final String PASSWORD = "12345";
    public static final int USER_TYPE_ID = 1;
    public static final String TIMEZONE = "GMT-4";
    public static final String REGISTRATION_CODE = "HIEN-API";
    public static final String EDMODO_SCHOOL_REF = "bd03e53e4801837807c5e023ffa816d9";

    static final String m_prefix = "PS";
    static final String m_version = "A";

    /*

```

```

* Transfer user from LTI to ASSISTments Token for Dao is currently the LTI
* user's ID
*/
public static ArrayList<String> transferUser(ToolProvider toolProvider,
        String outside_access_token,
        String currently_loggin_outside_user_token) {
    String userRef = "";
    String accessToken = "";
    ArrayList<String> assistAccount = new ArrayList<String>();

    String partnerExternalRef = toolProvider.getUser().getId();

    ExternalUser externalUser = new ExternalUser(PARTNER_REFERENCE);
    if (toolProvider.getUser().getId()
        .equals(currently_loggin_outside_user_token))
        externalUser.setUser_connector_token(outside_access_token);
    else
        externalUser.setUser_connector_token("");

    externalUser.setPartner_external_reference(partnerExternalRef);
    externalUser.setExternal_refernce_type_id(USER_TYPE_ID);

    org.assistments.connector.domain.User assistUser = getUserInfo(toolProvider);

    ExternalUserDAO userDAO = new ExternalUserDAO(PARTNER_REFERENCE);
    if (!userDAO.isUserExist(partnerExternalRef)) {
        // create new User
        Response r = UserController.createUser(assistUser,
            Utility.PARTNER_REFERENCE);
        if (r.getStatusCode() == 201) {
            JsonElement jElement = new JsonParser().parse(r.getContent());
            JsonObject jsonObject = jElement.getAsJsonObject();
            userRef = jsonObject.get("user").getAsString();
        }
        if (userRef == null || userRef.equals("")) {
            return null;
        }
        externalUser.setExternal_refernce(userRef);

        accessToken = createAccessToken(userRef);

        if (accessToken == null) {
            return null;
        }
        externalUser.setUser_access_token(accessToken);

        // update db
        userDAO.addNewUser(externalUser);
    } else {
        externalUser = userDAO.findByPartnerExternalRef(partnerExternalRef);

        if(externalUser.getUser_access_token() == null || externalUser.getUser_access_token() == "") {
            String tmp = createAccessToken(externalUser.getExternal_refernce());
            externalUser.setExternal_refernce(tmp);
        }
        // need to update outside access_token because change frequently
        userDAO.update(externalUser);
    }
}

```

```

        assistAccount.add(externalUser.getExternal_refernce());
        assistAccount.add(externalUser.getUser_access_token());

        return assistAccount;
    }
    /**
     * This method will check if class already exists. If not, a new class will be created and return class ref.
     * Otherwise, it will return class ref from partner_external_references table
     *
     * @param contextTitle -- class name from outsider
     * @param contextID -- class id from outsider
     * @param partnerRef
     * @param onBehalf
     * @return class reference from ASSISTments
     */
    public static String createClass(String contextTitle, String contextID, String partnerRef,
        String onBehalf) {
        String classRef = "";

        try {
            //TODO :
            String partnerExternalRef = contextTitle + contextID;
            ExternalStudentClass externalClass = new ExternalStudentClass(partnerRef);
            externalClass.setPartner_reference(PARTNER_REFERENCE);
            externalClass.setExternal_refernce_type_id(CLASS_SECTION);
            externalClass.setPartner_external_reference(partnerExternalRef);
            ExternalStudentClassDAO classDAO = new ExternalStudentClassDAO(partnerRef);
            if (!classDAO.isClassExist(partnerExternalRef)) {
                // prepare data
                StudentClass assisTmentsClass = new StudentClass(contextTitle,
                    Utility.COURSE_NUMBER, Utility.SECTION_NUMBER);
                Response r = StudentClassController.createStudentClass(assisTmentsClass, partnerRef,
onBehalf);

                if(r.getStatusCode() == 201) {
                    classRef = parseClassJson(r.getContent());
                }

                // insert new school into db
                externalClass.setExternal_refernce(classRef);
                externalClass.setUser_access_token(onBehalf); // who create the

                // class
                classDAO.addNewClass(externalClass);

            } else {
                externalClass = classDAO.findByPartnerExternalRef(partnerExternalRef);
                classRef = externalClass.getExternal_refernce();
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return classRef;
    }
}

public static String fetchClassRef(String contextID, String contextTitle, String onBehalf) {
    String classRef = "";

```

```

    try {

        ExternalStudentClassDAO classDAO = new ExternalStudentClassDAO(PARTNER_REFERENCE);
        String partnerExternalRef = contextTitle + contextID;
        if (!classDAO.isClassExist(partnerExternalRef)) {
            return null;
        } else {
            ExternalStudentClass studentClass =
classDAO.findByPartnerExternalRef(partnerExternalRef);
            classRef = studentClass.getExternal_refernce();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return classRef;
}

public static String createAssignment(String appld, String classRef, String onBehalf) {
    String assignmentRef = "";

    try {

        ExternalAssignment assignment = new ExternalAssignment(PARTNER_REFERENCE);
        assignment.setExternal_refernce_type_id(Constants.EXTERNAL_ASSIGNMENT_TYPE_ID);
        //TODO: I'm not sure this is the best way to set partner external reference. (Duplicate appld in the
same class) -- by Siyuan
        String partnerExternalRef = appld + classRef;
        assignment.setPartner_external_reference(partnerExternalRef);

        ExternalAssignmentDAO assignmentDAO = new ExternalAssignmentDAO(PARTNER_REFERENCE);
        if (!assignmentDAO.isAssignmentExist(partnerExternalRef)) {
            Response r = AssignmentController.createAssignment(appld, classRef,
Utility.PARTNER_REFERENCE, onBehalf);
            if(r.getStatusCode() == 201) {
                assignmentRef = parseAssignmentJson(r.getContent());
                assignment.setExternal_refernce(assignmentRef);
                assignmentDAO.addNewAssignment(assignment);
            }
        }
        else {
            assignment = assignmentDAO.findByPartnerExternalRef(partnerExternalRef);
            assignmentRef = assignment.getExternal_refernce();
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return assignmentRef;
}

private static String parseAssignmentJson(String assingmentRef) {
    JsonElement jElement = new JsonParser().parse(assingmentRef);
    JsonObject jObject = jElement.getAsJsonObject();
    String assignmentRef = jObject.get("assignment").toString(); // "d15daef3972c4a7528afd6f97e9f536a"
    assignmentRef = assignmentRef.substring(1, assignmentRef.length()-1);
    return assignmentRef;
}

```

```

protected static org.assistments.connector.domain.User getUserInfo(
    ToolProvider toolProvider) {
    org.assistments.connector.domain.User user = new org.assistments.connector.domain.User();
    User ltiUser = toolProvider.getUser();
    // principal or proxy by TEACHER or STUDENT
    if (ltiUser.isAdmin() || ltiUser.isStaff()) {
        user.setUserType(Constants.PRINCIPAL);
        user.setPassword(Utility.PASSWORD);
        user.setFirstName(ltiUser.getFirstname());
        user.setLastName(ltiUser.getlastname());

        // create a unique timestamp so email address and username is unique
        Long time = uniqueCurrentTimeMS();
        String email = time.toString() + "@lti.com";
        user.setEmail(email);
        user.setUsername(email);
        user.setDisplayName(ltiUser.getFullname());
        user.setTimeZone(Utility.TIMEZONE);
        user.setRegistrationCode(Utility.REGISTRATION_CODE);
    } else if (ltiUser.isLearner()) {
    } else {
        user.setUserType(Constants.PROXY);
        user.setPassword(Utility.PASSWORD);
        user.setFirstName(ltiUser.getFirstname());
        user.setLastName(ltiUser.getlastname());
        user.setEmail(ltiUser.getEmail());
        Long time = uniqueCurrentTimeMS();
        String email = time.toString() + "@lti.com";
        user.setEmail(email);
        user.setUsername(ltiUser.getFullname());
        user.setDisplayName(ltiUser.getFullname());
        user.setTimeZone(Utility.TIMEZONE);
        user.setRegistrationCode(Utility.REGISTRATION_CODE);
    }

    return user;
}

public static String getUserPageURL(String userType, String partnerRef, String onBehalf, String appld) {

    String addressToGo = "";

    String onFailure = "www.assistments.org";
    String onExit = "http://www.assistments.org:8080/connector/LTIGradeRequest";
    String onExit = Constants.CONNECTOR_URL + "LTIGradeRequest";
    String from = "LTI";

    if (userType.equals(Constants.TEACHER_ROLE)) {
        String onSuccess = Constants.TEACHER_PAGE+"/"+appld+"?from="+from;
        try {
            onSuccess = URLEncoder.encode(onSuccess, "UTF-8");
        } catch (Exception e) {
            e.printStackTrace();
        }
        addressToGo =
String.format("%1$s?partner=%2$s&access=%3$s&on_success=%4$s&on_failure=%5$s", Constants.LOGIN_URL, partnerRef,
onBehalf, onSuccess, onFailure);

```

```

        }else if (userType.equals(Constants.STUDENT_ROLE)) {
//            onExit = Constants.STUDENT_PAGE+"/"+appld+"?onExit=null&partner_id=3&from="+from;

            String onSuccess =
Constants.STUDENT_PAGE+"/"+appld+"?onExit="+onExit+"&partner_id=3&from="+from;
            try {
                onSuccess = URLEncoder.encode(onSuccess, "UTF-8");
            } catch (Exception e) {
                e.printStackTrace();
            }
            addressToGo =
String.format("%1$s?partner=%2$s&access=%3$s&on_success=%4$s&on_failure=%5$s", Constants.LOGIN_URL, partnerRef,
onBehalf, onSuccess, onFailure);
        }
        return addressToGo;
    }

/**
 * Create access token inside ASSISTments for this user based on user external reference
 * Before calling this function, make sure the user does not have access token
 * @param userRef -- use external reference inside ASSISTments
 * @return access token inside ASSISTments
 */
private static String createAccessToken(String userRef) {
//        String accessToken="";
//            ASSISTmentsRequest assistmentsRequest = new ASSISTmentsRequest();
//            assist_access_token = assistmentsRequest.createAccessToken(userRef);
Response r = UserController.creatAccessToken(userRef, Utility.PARTNER_REFERENCE);
//TODO: Check r.httpcode first
JsonElement jElement = new JsonParser().parse(r.getContent());
JsonObject jobject = jElement.getAsJsonObject();
accessToken = jobject.get("access").getAsString();
return accessToken;
}

private static String parseClassJson(String classRefJson) {
    JsonElement jElement = new JsonParser().parse(classRefJson);
    JsonObject jobject = jElement.getAsJsonObject();
    String classRef = jobject.get("class").toString(); // "d15daef3972c4a7528afd6f97e9f536a"
    classRef = classRef.substring(1, classRef.length() - 1);
    return classRef;
}

private static final AtomicLong LAST_TIME_MS = new AtomicLong();
public static long uniqueCurrentTimeMS() {
    long now = System.currentTimeMillis();
    while(true) {
        long lastTime = LAST_TIME_MS.get();
        if (lastTime >= now)
            now = lastTime+1;
        if (LAST_TIME_MS.compareAndSet(lastTime, now))
            return now;
    }
}

/**
 * Convert encoded problem set string to problem set id
 * @param psString -- encoded problem set string (PSxxxx)

```



```

        * @return problem set number id (a String)
        */
        public static String decodeProblemSetString(String psString) {
if (psString.isEmpty()) {
    return null;
}

// decode prefix
if (!psString.substring(0, m_prefix.length()).equalsIgnoreCase(m_prefix)) {
    return null;
}

// decode version
if (!psString.substring(m_prefix.length(), m_prefix.length() + m_version.length()).equalsIgnoreCase(m_version)) {
    return null;
}

// decode problem id
String code = "abcdefghijklmnopqrstuvwxyz23456789";
int decodedId = 0;
String psStringLowerCase = psString.toLowerCase();
for (int i = m_prefix.length() + m_version.length(); i < psStringLowerCase.length(); i++) {
    char c = psStringLowerCase.charAt(i);
    int oldValue = decodedId;
    decodedId = decodedId * code.length() + code.indexOf(c);
    if (decodedId < oldValue) {
        throw new RuntimeException("Overflow decoded id");
    }
}
return new Integer(decodedId).toString();
}
}
}

```

## Changes to the Ruby packages in `as_core` were made to support Ruby 1.8.6:

### **Oauth-0.4.7 -> Net\_http.rb**

```

def oauth_body_hash_required?
  request_body_permitted? #&& !content_type.to_s.downcase.includes?("application/x-www-form-urlencoded")
end

```

### **Actionpack-1.13.3 -> Cgi\_process.rb**

```

def body
  return @env['RAW_POST_DATA']
end

def url
  page_url = "http"
  if (@env['HTTPS'] == "on")
    page_url += "s"
  end
  page_url += "://"
  if (@env['SERVER_PORT'] != "80")
    page_url += @env['SERVER_NAME'] + ":" + @env['SERVER_PORT'] + @env['REQUEST_URI']
  else
    page_url += @env['SERVER_NAME'] + @env['REQUEST_URI']
  end
  return page_url
end

```

## Changes to the Ruby source in ast for LTI and Partial Credit:

### App\Controlllers\Build\Sequence\_controller.rb

```
def createlti
  @is_a_part_of_existing_problem_set = params.has_key?('section_id')
  @is_copy = params.has_key?('is_copy')
  if (params.has_key?(:commit))
    #We are saving the link
    title = params["title"]
    launchURL = params["launchurl"]
    consumerKey = params["consumerkey"]
    sharedSecret = params["secret"]
    ltiParams = params["cusparams"]

    if !(launchURL =~ URI::regexp)
      flash[:warning] = "Invalid URL! Please make sure a valid URL is entered (include http:// or https://)"
      return
    end

    new_assistments = Array.new
    new_assistment_tutor_modes = Array.new

    if (@is_a_part_of_existing_problem_set && !@is_copy)
      #We are saving a new LTI link
      section = Section.find(params[:section_id])

      section.sequence[:parameters]["lti_title"] = title
      section.sequence[:parameters]["lti_launchurl"] = launchURL
      section.sequence[:parameters]["lti_key"] = consumerKey
      section.sequence[:parameters]["lti_secret"] = sharedSecret
      section.sequence[:parameters]["lti_params"] = ltiParams
      section.sequence[:parameters]["is_link"] = "true"
      section.sequence.save
      redirect_to(:action => :index)
    else
      #We are saving an existing LTI link
      sequence = Sequence.create_from_hash({
        'name' => title,
        'assistments' => new_assistments,
        'tutor_modes' => new_assistment_tutor_modes
      })
      sequence.content_creators << current_user.content_creator
      sequence[:parameters]["lti_title"] = title
      sequence[:parameters]["lti_launchurl"] = launchURL
      sequence[:parameters]["lti_key"] = consumerKey
      sequence[:parameters]["lti_secret"] = sharedSecret
      sequence[:parameters]["lti_params"] = ltiParams
      sequence[:parameters]["is_link"] = "true"
      sequence.save
      redirect_to(:action => :index)
    end
  elsif @is_a_part_of_existing_problem_set
    #We are going to edit an existing LTI link
    section = Section.find(params[:section_id])
    @title = section.sequence[:parameters]["lti_title"]
    @launchURL = section.sequence[:parameters]["lti_launchurl"]
    @consumerKey = section.sequence[:parameters]["lti_key"]
  end
end
```

```

        @sharedSecret = section.sequence[:parameters]["lti_secret"]
        @ltiParams = section.sequence[:parameters]["lti_params"]
    end
end

def show
  ...
  if (@sequence.lti_link?)
    redirect_to(:controller => :sequence, :action => :createlti, :section_id => @sequence.head_section_id)
    return
  end
  ...
end

def duplicate
  ...
  sequence = Sequence.find_by_id(sequence_id, :include => [:base_section, :sections, :content_creators])
  if sequence.lti_link?
    redirect_to(:controller => :sequence, :action => :createlti, :section_id =>
sequence.head_section_id, :is_copy => 'true')
    return
  end
  ...
End

```

#### **App\Controllors\api2\api\_helper\_controller.rb**

```

def grade_passback
  req = IMS::LTI::OutcomeRequest.from_post_request(request)
  sourcedid = req.lis_result_sourcedid.to_s

  sourceParams = sourcedid.split("::")
  if(sourceParams.size!=4)
    render :text => "Failed to perform LTI Grade Return"
    return
  end
  consumerKey = sourceParams[0]
  secret = sourceParams[1]
  userID = sourceParams[2]
  assignmentID = sourceParams[3]

  consumer = IMS::LTI::ToolConsumer.new(consumerKey, secret)

  if consumer.valid_request?(request)
    if consumer.request_oauth_timestamp.to_i - Time.now.utc.to_i > 60*60
      render :text => "Failed to perform LTI Grade Return"
      return
    end
    # todo: check if nonce was used in a repeat, requires caching or db

    res = IMS::LTI::OutcomeResponse.new
    res.message_ref_identifier = req.message_identifier
    res.operation = req.operation
    res.code_major = 'success'
    res.severity = 'status'

    assignment_log = TutorHelper.get_or_create_assignment_log(userID, assignmentID, "ClassAssignment")
    if req.replace_request?
      res.description = "Your old score has been replaced with #{req.score}"
    end
  end
end

```

```

        assignment_log.variables["lti_grade"] = req.score.to_s
      elsif req.read_request?
        res.description = "Your score is #{assignment_log.variables["lti_grade"]}"
        res.score = assignment_log.variables["lti_grade"]
      elsif req.delete_request?
        res.description = "Your score has been cleared"
        assignment_log.variables["lti_grade"] = ""
      else
        res.code_major = 'unsupported'
        res.severity = 'status'
        res.description = "#{req.operation} is not supported"
      end

      response.headers["Content-Type"] = 'text/xml'
      res.generate_response_xml
      TutorHelper.set_assignment_log(userID, assignmentID, assignment_log, "ClassAssignment")
      render :text => "Successfully stored LTI Grade"
    else
      render :text => "Failed to perform LTI Grade Return"
      return
    end
  end
end
end

```

#### **App\Controlllers\Teacher\folder\_controller.rb**

```
Def assign_folder_item
```

```
...
```

```
#Make sure we didn't redirect to a link, to avoid a redirect/render issue
```

```
  if @lti_link.nil?
```

```
    render :template => template_string, :type => :rjs, :layout => false
```

```
  end
```

```
...
```

```
end
```

```
def assign_folder_item_to_class
```

```
...
```

```
@lti_link = nil
```

```
  if @folder_item.item.class.to_s == 'Folder'
```

```
    flash[:notice] = 'Folder Successfully Assigned to ' + @student_class.name if flash[:warning].blank?
```

```
  else
```

```
    #Launch if LTI Link
```

```
    if !@assigned_item.nil? and @assigned_item.lti_link?
```

```
      flash[:warning] = 'External assignment assigned! Please wait to be redirected to the external
```

```
assignment...'
```

```
      render :update do |page|
```

```
        page.redirect_to :controller => "/tutor/student_class", :action =>
```

```
"tool_launch", :assignmentID => @assigned_item.id
```

```
      end
```

```
      @lti_link = true
```

```
      return
```

```
    end
```

```
    flash[:notice] = 'Problem Set Successfully Assigned to ' + @student_class.name if flash[:warning].blank?
```

```
  end
```

```
...
```

```
End
```

```
def create_class_assignment valid_sequence_ids
```

```
...
```

```
#Launch if LTI Link
```

```

        if !ca.nil? and ca.lti_link?
            flash[:warning] = 'External assignment assigned! Please wait to be redirected to the external
assignment...'
            render :update do |page|
                page.redirect_to :controller => "/tutor/student_class", :action =>
"tool_launch", :assignmentID => ca.id
            end
            @lti_link = true
            return
        end
    ...
End

```

```

Def assign_to_sections_or_individuals
...
#Launch if LTI
    @lti_link = nil
    if !@assigned_item.nil? and @assigned_item.lti_link?
        flash[:warning] = 'External assignment assigned! Please wait to be redirected to the external
assignment...'
        render :update do |page|
            page.redirect_to :controller => "/tutor/student_class", :action =>
"tool_launch", :assignmentID => @assigned_item.id
        end
        @lti_link = true
        return
    end
...
End

```

#### App\Controllers\Tutor\Student\_class\_controller.rb

```

def tool_launch
    assignmentID = params["assignmentID"]
    if(assignmentID.nil?)
        flash[:warning] = "No assignment found! Please contact assistments-help@wpi.edu for support."
        return
    end
    assignment = ClassAssignment.find(assignmentID)
    title = assignment.sequence.parameters["lti_title"]
    launchURL = assignment.sequence.parameters["lti_launchurl"]
    consumerKey = assignment.sequence.parameters["lti_key"]
    sharedSecret = assignment.sequence.parameters["lti_secret"]
    ltiParams = assignment.sequence.parameters["lti_params"]

    if !(launchURL =~ URI::regexp)
        flash[:warning] = "An external assignment was launched with an invalid URL! Please contact assistments-
help@wpi.edu
        for support, and please provide the following:
        <br>Assignment: " + assignmentID +
        "<br>URL: " + launchURL
    end
    return
end

tc = IMS::LTI::ToolConfig.new(:title => title, :launch_url => launchURL )

#Add custom parameters
listParam = ltiParams.split(/\r?\n/)
if(!listParam.nil?)

```

```

        listParam.each do |i|
            splitParam = i.split("=")
            if(!splitParam.nil? && splitParam.size == 2)
                tc.set_custom_param(splitParam[0].downcase.gsub(/\/W+/, '_'), splitParam[1])
            end
        end
    end
end

@consumer = IMS::LTI::ToolConsumer.new(consumerKey, sharedSecret)
@consumer.set_config(tc)

# Only this first one is required, the rest are recommended
# Resource link ID will be unique ID combined of the assignment and user's e-mail
@consumer.resource_link_id = assignmentID.to_s + current_user.email
@consumer.resource_link_title = assignment.name
@consumer.launch_presentation_return_url = ""
@consumer.lis_person_name_given = current_user.user_detail.first_name
@consumer.lis_person_name_family = current_user.user_detail.last_name
@consumer.lis_person_name_full = current_user.user_detail.name
@consumer.lis_person_contact_email_primary = current_user.email
@consumer.user_id = current_user.id

#Add roles
String allRoles = ""
if(params["studentlaunch"].nil?)
    if(current_user.is_administrator?)
        allRoles = "Administrator"
    elsif(current_user.is_staff?)
        allRoles = "Instructor"
    else
        allRoles = "Learner"
    end
else
    allRoles = "Learner"
end

# Context_id is the assignment-id specifying its unique launch location
@consumer.roles = allRoles
@consumer.context_id = assignmentID.to_s
@consumer.context_title = title
@consumer.context_label = "ASST"
@consumer.tool_consumer_instance_name = "ASSISTments"
@consumer.tool_consumer_instance_guid = "www.assistments.org"
@consumer.tool_consumer_instance_url = "http://www.assistments.org"
@consumer.tool_consumer_instance_description = "ASSISTments"
@consumer.tool_consumer_instance_contact_email = "assistments-help@wpi.edu"

# URL for returning grades, along with the necessary parameters for authentication
@consumer.lis_outcome_service_url = request.protocol + request.host_with_port + "/api2_helper/grade_passback"
@consumer.lis_result_sourcedid = consumerKey + ":@" + sharedSecret + ":@" + current_user.id.to_s + ":@" +
assignmentID.to_s

#If false, will take to debug page
@autolaunch = true

End

App\Helpers\application_helper.rb
def render_lti_link class_assignment, name, color

```

```

url = "/tutor/student_class/tool_launch?assignmentID=#{class_assignment.id}"
return link_to(name, url, :popup => ['External Assignment', 'height=600, width=600'], :style => "#{color}")

end

def render_lti_link_student class_assignment, name, color
  url = "/tutor/student_class/tool_launch?assignmentID=#{class_assignment.id}&studentlaunch=true"
  return link_to(name, url, :popup => ['External Assignment', 'height=600, width=600'], :style => "#{color}")

end

def render_lti_grade class_assignment
  assignment_log = TutorHelper.get_or_create_assignment_log(current_user.id, class_assignment.id,
"ClassAssignment")
  if !assignment_log.variables["lti_grade"].nil?
    return "Grade: " + assignment_log.variables["lti_grade"]
  end
end
end

```

#### App\Models\sequence.rb

```

def lti_link?
  return false if self.parameters.blank? || self.parameters["is_link"].nil?
  return true
end

def self.lti_link? sequences
  return true unless sequences.select{|seq| seq.lti_link?}.blank?
  return false
end

```

#### App\Models\class\_assignment.rb

```

def lti_link?
  return false if self.sequence.blank?
  return self.sequence.lti_link?
end

```

#### App\Views\Build\Sequence\index.rhtml

```

...
<% if !@teacher_settings["show_builder_lti_link"].nil? and
@teacher_settings["show_builder_lti_link"].setting_value=="true" %>
  <td width="300"><%= link_to 'Create Your Own LTI Link', {:controller => "build/sequence", :action
=> :createlti}, :class => :new %></td>
  <%end%>
...

```

#### App\Views\Build\Sequence\createlti.rhtml

```

<%= javascript_include_tag "quickbuilder" %>
<%= javascript_include_tag "report.js" %>

<script src="/javascripts/jquery-1.8.3.js?" type="text/javascript"></script>
<script src="/javascripts/jquery-ui.js?" type="text/javascript"></script>
<link rel="stylesheet" href="/stylesheets/build/jquery-ui.css" />

<script type="text/javascript">
  var $j = jQuery.noConflict();
  $j(function() {
    $j('.container-5').tabs();
  });

```

```

    });
</script>
<script type="text/javascript">

</script>

<div class="warning_message" style="font-family:'Arial'; font-size:14px">
  <b>LTI is used to connect assignments to ASSISTments.<br/>
  ASSISTments will share the launched username and e-mail with the tool.<br/>
  Please make sure to enter a valid URL, along with any extra necessary information.<br/>
  Custom parameters and their values are split by '=', each paramter must be on a newline. Ex:<br/>
  param1=value<br/>param2=value2<br/><br/>
  Invalid parameters will be ignored.
</b>
</div>

```

```

<div id='field' style="padding: 4px; border: solid 2px black; ">
  <% form_tag((url_for :action => :createlti)) do %>

    <%if @is_a_part_of_existing_problem_set%>
      <%= hidden_field_tag("section_id", params["section_id"])%>
    <%end%>
    <%if @is_copy%>
      <%= hidden_field_tag("is_copy", params["is_copy"])%>
    <%end%>
    <div>
      <label for="Title">Title:</label>
    <div>
      <%= text_field_tag 'title', @title, {:size => "50"} %>
    </div>
    <div>
      <label for="Launch URL">Launch URL:</label>
      <%= text_field_tag 'launchurl', @launchURL, {:size => "43"} %>
    </div>
    <div>
      <label for="Consumer Key">Consumer Key:</label>
      <%= text_field_tag 'consumerkey', @consumerKey, {:size => "41"} %>
    </div>
    <div>
      <label for="Shared Secret">Shared Secret:</label>
      <%= text_field_tag 'secret', @sharedSecret, {:size => "41"} %>
    </div>
    <div>
      <label for="Custom Parameters">Custom Parameters:</label>
      <%= text_area_tag 'cusparams', @ltiParam, {:size => "35x6"} %>
    </div>
    <%= submit_tag 'Create LTI', {:class => "submit", :style=>"margin-left:0px"} %>

    <% end %>
  </div>

```

```

<div id="dialogBox" class="dialog">
  <%= render :partial => 'dialog_box' %>
</div>

```

App\Views\Folder\\_assign\_form.rhtml



```

...
if folder_item.item.sequence.lti_link?
  function_str += " window.open('http://www.assistments.org/teacher', 'ASSISTments', 'height=1200,width=1200');"
end
...

App\Views\Teacher\Folder\_class_assignment.rhtml
...
<% unless @assignment.lti_link? %>
  <%= link_to('Report', :controller => 'teacher/report', :action => 'item_level', :assignment_id => @assignment.id, :id =>
@student_class.id) %>
  <% else %>
    <%= link_to('Report', :controller => 'teacher/report', :action =>
'assignment_level', :assignment_id => @assignment.id, :id => @student_class.id) %>
  <% end %>
... ..
<% if type == 'sequence' and !@assignment.lti_link? %>
  ~
  <%= render_print_link(:assignment, @assignment.id, "View Problems", :html_options => {:style => 'color: green'}) %>
  ~
  <%= render_preview_link(:assignment, @assignment, "Test Drive", :style => 'color: green') %>
<% elsif type == 'sequence' and @assignment.lti_link? %>
  ~
  <%= render_lti_link(@assignment, "Test Drive", "color: green;") %>
<% end %>
... ..
<% if type == 'sequence' and !@assignment.lti_link? %>
  ~
  <%= render_print_link(:assignment, @assignment.id, "View Problems", :html_options => {:style => 'color: green'}) %>
  ~
  <%= render_preview_link(:assignment, @assignment, "Test Drive", :style => 'color: green') %>
<% elsif type == 'sequence' and @assignment.lti_link? %>
  ~
  <%= render_lti_link(@assignment, "Test Drive", "color: green;") %>
<% end %>
...
App\Views\Teacher\Folder\_class_assignment_dates.rhtml
<% unless assignment.lti_link? %>
  <span style="color:red;font-weight:normal">You have not set a due date for this assignment!</span><br />
<% end %>

App\Views\Teacher\student_class\folder\_sequence_actions.rhtml
<% if !sequence.lti_link? %>
  ~
  <%= render_print_link(:sequence, sequence.id, "View Problems", :mode => 'test', :html_options => {:style => 'color:
green'}) %>
  ~
  <%= render_preview_link(:sequence, sequence, "Test Drive", :style => 'color: green') %>
<% end %>

App\Views\Tutor\Folder\_lti_class_assignment.rhtml
<%
# View for rendering regular class assignments
# assignments that are not ARRS.
raise "class_assignment nil in /tutor/class_assignment/_reg_class_assignment.rhtml" if class_assignment.blank?

# !!!IMPORTANT!!! : This outer div is required for folders to display nicely
-%>

```

```

<div id='folder_item_folder_item.id ' class="folder_item">
  <div id="folder_item_folder_item.id_notice"></div>
  <% name = class_assignment.title
  status = @aggregate[:progresses][class_assignment.id] -%>

  <div class="class_assignment item <%= status.to_s %>" id="<%= dom_id(class_assignment) %>"

  <!-- This code deals with the "name" part of each class assignment -->
  <table style="width: 100%;">
    <tr>
      <td>
        <b><%= render_lti_link_student class_assignment, name, "color: #9900FF;" %></b>
      </td>
      <td style="text-align: right;">
        <%= render_lti_grade class_assignment %>
      </td>
    </tr>
  </table>
</div>
</div>

```

#### App\Views\Tutor\Folder\\_class\_assignment.rhtml

```

...
<% if class_assignment.lti_link? -%>
  <%= render( :partial => 'lti_class_assignment',
  :locals => {
    :class_assignment => class_assignment,
    :folder_item => folder_item,
    :parameters => parameters
  })%>
...

```

#### App\Views\Tutor\student\_class\tool\_launch.rhtml

```

<%= javascript_include_tag "quickbuilder" %>
<%= javascript_include_tag "report.js" %>

<script src="/javascripts/jquery-1.8.3.js?" type="text/javascript"></script>
<script src="/javascripts/jquery-ui.js?" type="text/javascript"></script>
<link rel="stylesheet" href="/stylesheets/build/jquery-ui.css" />

<script type="text/javascript">
  var $j = jQuery.noConflict();
  $j(function() {
    $j('.container-5').tabs();
  });
</script>

<% if !@consumer.nil? %>
  <div class="warning_message" style="font-family:'Arial'; font-size:14px">
    Redirecting you to an external assignment.<br/>
    Please wait...
  </div>
  <div id="ltiLaunchFormSubmitArea">
    <!-- For seeing all the parameters, delete these comment lines for debugging
    <table>
      <tr>
        <td colspan="2">Launch Parameters</td>

```

```

        </tr>
        <% @consumer.generate_launch_data.each_pair do |key, val| %>
          <tr>
            <td><%= key %></td>
            <td><%= val %></td>
          </tr>
        <% end %>
      </table>
    !-->
    <form action="<%= @consumer.launch_url %>" name="ItiLaunchForm" id="ItiLaunchForm" method="post"
    encType="application/x-www-form-urlencoded">
      <% @consumer.generate_launch_data.each_pair do |key, val| %>
        <input type="hidden" name="<%= key %>" value="<%= val %>"/>
      <% end %>
      <button type="submit">Launch the tool</button>
    </form>

  </div>
  <% if @autolaunch %>
    <script language="javascript">
      document.getElementById("ItiLaunchFormSubmitArea").style.display = "none";
      document.ItiLaunchForm.submit();
    </script>
  <% end %>
<% end %>

```

### Db\Migrate\330\_create\_show\_lti\_link\_in\_teacher\_setting.rb

```

class CreateCreditSettings < ActiveRecord::Migration
  def self.up
    ActiveRecord::Base.connection.execute("insert into default_settings (released_feature, in_use, invalidates, key_name,
    setting_value_format, setting_value, display_name, component, controller_path, settings_link_text, updated_at, description)
    values(true, true, false, 'credit_setting', 'Boolean', false, 'Partial Credit For Student Work', 'TEACHER',
    ", ", 'now()', '')")
    ActiveRecord::Base.connection.execute("update default_settings set display_position = id where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("insert into scope_settings (scope_id, default_setting_id, exposure) values
    (#{Scope::TEACHER}, (select id from default_settings where key_name = 'credit_setting') , 'public');")

    ActiveRecord::Base.connection.execute("insert into default_settings (released_feature, in_use, invalidates, key_name,
    setting_value_format, setting_value, display_name, component, controller_path, settings_link_text, updated_at, description)
    values(true, true, false, 'credit_attempts', 'NUMBER - ##', 3, 'Partial Credit For Student Work',
    'TEACHER', ", ", 'now()', '')")
    ActiveRecord::Base.connection.execute("update default_settings set display_position = id where key_name
    = 'credit_attempts'")
    ActiveRecord::Base.connection.execute("insert into scope_settings (scope_id, default_setting_id, exposure) values
    (#{Scope::TEACHER}, (select id from default_settings where key_name = 'credit_attempts') , 'public');")
  end

  def self.down
    ActiveRecord::Base.connection.execute("delete from scope_settings where default_setting_id in (select id from
    default_settings where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("delete from default_settings where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("delete from scope_settings where default_setting_id in (select id from
    default_settings where key_name = 'credit_attempts'")
    ActiveRecord::Base.connection.execute("delete from default_settings where key_name = 'credit_attempts'")
  end
end

```

### App\Controllers\Teacher\report\_controller.rb

```

def partial_credit_from_log log
  if log.nil? or log.user.nil? or log.problem.nil?
    return 0.0
  end
  student = log.user
  problem = log.problem.id
  if @student_partial_credit.nil?
    @student_partial_credit = Hash.new
  end
  if !@student_partial_credit[student.to_s + '/' + problem.to_s].nil?
    return @student_partial_credit[student.to_s + '/' + problem.to_s]
  else
    score = 0.0
    curr_score = 1.0
    # Use 1 maximum attempts if the credit setting is nil or turned off
    maxAttempts = (@student_class.credit_setting.nil? or @student_class.credit_setting==false ? 1 :
@student_class.credit_attempts.to_i)
    if !log.nil? and log.complete? and log.correct.to_i == 1
      score += 1.0
    elsif !log.nil? and log.complete? and log.correct.to_i == 0 and log.attempt_count > 0
      #Calculate credit based on whether or not it's multiple choice
      if log.problem.multiple_choice?
        curr_score = 1.0 - log.attempt_count * (1.0/(Answer.count_by_sql("SELECT COUNT(*)
from Answers WHERE problem_id=#{problem}"))) - 1.0))
        (curr_score < 0) ? curr_score = 0 : curr_score
      end
      score = 1.0 - (log.hint_count+log.attempt_count-1.0)*(1.0/maxAttempts)
      (score < 0) ? score = 0 : score
      score = [score, curr_score].min
    end
    @student_partial_credit[student.to_s + '/' + problem.to_s] = score
  end
  return score
end
end

def assignment_level_progress
  # getting progresses for @students and @assignments
  student_user_ids = @students.collect{|s| s.user.id}
  assignment_ids = @assignments.collect{|a| a.id}
  @assignment_logs = AssignmentLog.find(:all, :conditions => ["assignment_id in (?) and user_id in (?) and
assignment_type = 'ClassAssignment'", assignment_ids, student_user_ids])

  #building the progress hash now
  @progresses = Hash.new
  @assignment_logs.each do |progress|
    @progresses[progress.user_id] = Hash.new if @progresses[progress.user_id].nil?
    @progresses[progress.user_id][progress.assignment_id] = Hash.new if
@progresses[progress.user_id][progress.assignment_id].nil?
    @progresses[progress.user_id][progress.assignment_id] = progress
  end

  # calculating averages
  @average = Hash.new

  assistments = ProblemLog.find_by_sql(
    ["select l.* "+
    "from problem_logs l, class_assignments c "+

```

```

"where c.id in (?) and l.assignment_id = c.id and c.student_class_id = ? and l.end_time is not null and
l.assignment_type = 'ClassAssignment' "+
"order by l.assignment_id, l.end_time desc", assignment_ids, @student_class_id]
)
assistments.each do |assistment|

  next unless assistment.original == 1
  @average[assistment.user_id] = Hash.new if @average[assistment.user_id].nil?
  if @average[assistment.user_id][assistment.assignment_id].nil?
    @average[assistment.user_id][assistment.assignment_id] = Hash.new
    @average[assistment.user_id][assistment.assignment_id]['total_done'] = 0
    @average[assistment.user_id][assistment.assignment_id]['total_correct'] = 0
  end
  @average[assistment.user_id][assistment.assignment_id]['total_done'] += 1 unless assistment.correct.nil?
  @average[assistment.user_id][assistment.assignment_id]['total_correct'] +=
partial_credit_from_log(assistment) unless assistment.correct.nil? #assistment.correct unless assistment.correct.nil?
end
@class_average = Hash.new
@number_of_started = Hash.new
@assignments.each do |assignment|
  @number_of_started[assignment.id] = @progresses.select{|k,v| !@progresses[k][assignment.id].nil?}.size
  if @class_average[assignment.id].nil?
    @class_average[assignment.id] = Hash.new
    @class_average[assignment.id]['average']=0
    @class_average[assignment.id]['complete']=0
  end

  @progresses.select{|k,v| !@progresses[k][assignment.id].nil? }.each do |student|
    @class_average[assignment.id]['complete'] += 1 if
@progresses[student[0]][assignment.id].complete?
    if assignment.lti_link?
      log = @assignment_logs.select{|x| assignment.id == x.assignment_id}.first
      unless log.nil?
        grade = log.variables["lti_grade"]
        if(grade.to_f <= 1)
          storedgrade = (grade.to_f*100)
        else
          storedgrade = grade.to_f
        end
        @class_average[assignment.id]['average'] += storedgrade
      end
    end
  else
    @class_average[assignment.id]['average']
+=(@average[student[0]][assignment.id]['total_correct']*100/@average[student[0]][assignment.id]['total_done']) unless
@average[student[0]].nil? or @average[student[0]][assignment.id].nil? or @average[student[0]][assignment.id]['total_done']
== 0
  end
end
@class_average[assignment.id]['average'] =
(@class_average[assignment.id]['average']/(@progresses.select{|k,v| !@progresses[k][assignment.id].nil?}.size)).to_i.to_s
unless (@progresses.select{|k,v| !@progresses[k][assignment.id].nil?}.size) ==0
  @class_average[assignment.id]['complete'] =
(@class_average[assignment.id]['complete']*100/@students.size).to_i.to_s unless @students.size == 0

end

end

end

```

```

def individual_student_level_progress
  @student_info = Hash.new
  assignment_ids = @assignments.collect{|a| a.id}
  @assignment_logs = AssignmentLog.find(:all, :conditions => ["assignment_id in (?) and user_id = (?) and
assignment_type = 'ClassAssignment'", assignment_ids, @student_user.id])

  #grades
  plogs = ProblemLog.find_by_sql(
    ["select l.assignment_id, count(l.assignment_id) as done, sum(correct) as correct "+
    "from problem_logs l, class_assignments c "+
    "where c.id in (?) and l.assignment_id = c.id and l.user_id = ? and l.end_time is not null and
l.assignment_type = 'ClassAssignment' "+
    "group by l.assignment_id", assignment_ids, @student_user.id]
  ).each{|plog|
    @student_info[plog.assignment_id] = Hash.new if @student_info[plog.assignment_id].nil?
    @student_info[plog.assignment_id]["done"] = plog.done.to_i
    @student_info[plog.assignment_id]["correct"] = plog.correct.to_i
    @student_info[plog.assignment_id]["percent"] = "#(((plog.correct.to_i * 100) / plog.done.to_i).to_i)%"
  }

  @assignments.each{|assignment|
    status = "not_started"
    days_late = ""
    last_worked_on = ""
    log = @assignment_logs.select{|x| assignment.id == x.assignment_id}.first

    #Calculate partial credit
    @student_info[assignment.id] = Hash.new if @student_info[assignment.id].nil?
    score = 0.0

    logCount = 0.0
    assignment.problem_logs.each{|aplog|
      score += partial_credit_from_log(aplog)
      logCount += 1.0 unless aplog.attempt_count < 1
    }
    @student_info[assignment.id]["percent"] = "#(((score * 100) / logCount).to_i)%" unless logCount==0
    unless log.nil?

      #days late
      if log.complete? and !assignment.due_date.nil? and assignment.due_date < log.end_time
        days = (log.end_time.to_date - assignment.due_date.to_date).to_i
        if days < 1
          days_late = "Less than 1 day late"
        elsif days == 1
          days_late = "1 day late"
        else
          days_late = "#{days} days late"
        end
      elsif !log.complete? and !assignment.due_date.nil? and assignment.due_date < Time.now
        days_late = "was due on #{assignment.due_date.strftime("%m-%d-%Y")}"
      end

      unless(log.last_worked_on.nil?)
        last_worked_on = log.last_worked_on.strftime("%m-%d-%Y")
        last_worked_on_link = log.last_worked_on.strftime("%Y-%m-%d")
      else
        last_worked_on = ""
        last_worked_on_link = ""
      end
    end
  }
end

```

```

        if log.complete?
            status = "correct"
        elsif log.in_progress?
            status = "in_progress"
        end
    end
end

@student_info[assignment.id]["days_late"] = days_late
@student_info[assignment.id]["last_worked_on"] = last_worked_on
@student_info[assignment.id]["last_worked_on_link"] = last_worked_on_link

@student_info[assignment.id]["status"] = status

if assignment.assignment_type_id == 1 #Normal class assignment
    if assignment.mastery_learning?
        display = "Skillbuilder"
    elsif assignment.lti_link?
        display = "External Assignment"
        unless log.nil?
            @student_info[assignment.id]["percent"] = log.variables["lti_grade"]
        end
    else
        @student_info[assignment.id]["percent"] = ""
    end
    @student_info[assignment.id]["done"] = ""
    @student_info[assignment.id]["correct"] = ""
else
    display = "Regular Problem Set"
end

elsif assignment.assignment_type_id == 6 #ARRS Relearning
    display = "Relearning Assignment"
elsif assignment.assignment_type_id == 7 #ARRS Reassessment
    display = "Reassessment Test"
else
    display = assignment.assignment_type.display_name
end

@student_info[assignment.id]["type_of_assignment"] = display

}
End

```

#### **App\helpers\teacher\report\_helper.rb**

```

def partial_credit student, problem
    if @student_partial_credit.nil?
        @student_partial_credit = Hash.new
    end
    if !@student_partial_credit[student.to_s + '/' + problem.to_s].nil?
        return @student_partial_credit[student.to_s + '/' + problem.to_s]
    else
        log = get_log student, problem
        score = 0.0
        curr_score = 1.0
        # Use 1 maximum attempts if the credit setting is nil or turned off
        maxAttempts = (@student_class.credit_setting.nil? or @student_class.credit_setting==false ? 1 :
@student_class.credit_attempts.to_i)
        if !log.nil? and log.complete? and log.correct.to_i == 1
            score += 1.0
        end
    end
end

```

```

        elsif !log.nil? and log.complete? and log.correct.to_i == 0 and log.attempt_count > 0
          #Calculate credit based on whether or not it's multiple choice
          if log.problem.multiple_choice?
            curr_score = 1.0 - log.attempt_count * (1.0/( (Answer.count_by_sql("SELECT
COUNT(*) from Answers WHERE problem_id=#{problem}") - 1.0))
            (curr_score < 0) ? curr_score = 0 : curr_score
          end
          score = 1.0 - (log.hint_count+log.attempt_count-1.0)*(1.0/maxAttempts)
          (score < 0) ? score = 0 : score
          score = [score, curr_score].min
        end
        @student_partial_credit[student.to_s + '/' + problem.to_s] = score
        return score
      end
    end

  end

  def get_problem_average problem
    return " if @problem_average[problem].nil? or @problem_average[problem]["total_done"] == 0
    #Partial Credit
    student_score = 0.0
    num_probs = 0.0
    @students.each do |student|
      student_score += partial_credit(student, problem)
      num_probs += 1
    end
    finalScore = (student_score / num_probs * 100.0).to_i
    return finalScore.to_s + '%'
    #(@problem_average[problem]["total_correct"].to_f * 100 /
@problem_average[problem]["total_done"].to_f).round().to_s + '%'
  end

  def get_student_average student
    key = student.user_id
    not_started = "
    return not_started if @student_average.nil?
    return not_started if @student_average[key].nil?

    # If they haven't done any regular problems, AND they have no
    # graded essays, we consider them "not started".
    if @student_average[key]["total_done"] == 0 and
    (@student_average[key]["graded_essay"].nil? or @student_average[key]["graded_essay"] == 0)
      return not_started
    end

    total_problems = get_number_of_non_open_response(@assignment_id.nil? ? @class_assignment_id : @assignment_id)

    #Add the number of graded essays to the total_problems
    total_problems += @student_average[key]["graded_essay"] if !@student_average[key]["graded_essay"].nil?

    return not_started if total_problems == 0
    #Partial Credit
    student_score = 0.0
    @problems.each do |problem|
      student_score += partial_credit(student, problem)
    end
    the_avg = (student_score / @student_average[key]["total_done"] * 100.0).to_i
    return the_avg.to_s + '%'
    #the_avg = (@student_average[key]["total_correct"].to_f * 100 / total_problems).round().to_s + '%'

```



```

#return the_avg
  End
def get_class_average
  total_done = 0
  total_correct = 0
  problems_in_assignment = get_number_of_non_open_response(@assignment_id.nil? ? @class_assignment_id :
@assignment_id)

  @student_average.each_pair{|id,student_hash|
    if student_hash["total_done"].to_i > 0
      total_done += @student_average[id]["total_done"]
      #total_done += problems_in_assignment
      #total_correct += (student_hash["total_correct"].to_f - student_hash["graded_essay_points"].to_f)
    end
  }

  #Partial credit
  class_score = 0.0
  @students.each do |student|
    @problems.each do |problem|
      class_score += partial_credit(student, problem)
    end
  end
  end
  return " unless total_done!=0
  finalScore = (class_score / total_done * 100.0).to_i
  return finalScore.to_s + '%'
  # return " if total_done==0
  # return (total_correct.to_f * 100 / total_done.to_f).round().to_s + '%'
  End

```

#### App\helpers\tutor\report\_helper.rb

```

def partial_credit student, problem
  if @student_partial_credit.nil?
    @student_partial_credit = Hash.new
  end
  if !@student_partial_credit[student.to_s + '/' + problem.to_s].nil?
    return @student_partial_credit[student.to_s + '/' + problem.to_s]
  else
    log = get_log student, problem
    score = 0.0
    curr_score = 1.0
    # Use 1 maximum attempts if the credit setting is nil or turned off
    maxAttempts = (@student_class.credit_setting.nil? or @student_class.credit_setting==false ? 1 :
@student_class.credit_attempts.to_i)
    if !log.nil? and log.complete? and log.correct.to_i == 1
      score += 1.0
    elsif !log.nil? and log.complete? and log.correct.to_i == 0 and log.attempt_count > 0
      #Calculate credit based on whether or not it's multiple choice
      if log.problem.multiple_choice?
        curr_score = 1.0 - log.attempt_count * (1.0/( (Answer.count_by_sql("SELECT
COUNT(*) from Answers WHERE problem_id=#{problem}"))) - 1.0))
        (curr_score < 0) ? curr_score = 0 : curr_score
      end
      score = 1.0 - (log.hint_count+log.attempt_count-1.0)*(1.0/maxAttempts)
      (score < 0) ? score = 0 : score
      score = [score, curr_score].min
    end
    @student_partial_credit[student.to_s + '/' + problem.to_s] = score
  end
  return score

```

```

        end
    end

    #Gets the class average for a particular problem.
    #Used in the student item report
    def get_problem_average problem
        if @assignment_information[:problem_average][problem].blank?
            return "N/A"
        else
            #If the problem is open response, we want a decimal answer scaled by the total possible
            if !@assignment_information[:problem_average][problem]["open_response_total"].blank? and
                @assignment_information[:problem_average][problem]["open_response_total"] > 0
                return 'N/A' if @assignment_information[:problem_average][problem]["total_done"] == 0
                return (((@assignment_information[:problem_average][problem]["total_correct"].to_f /
                    @assignment_information[:problem_average][problem]["total_done"]) *
                    @assignment_information[:problem_average][problem]["open_response_total"]) * 100).round() / 100.0).to_s
            else
                return 'N/A' if @assignment_information[:problem_average][problem]["total_done"] == 0
                #Partial Credit
                student_score = 0.0
                @student_class_students.each do |student|
                    student_score += partial_credit(student, problem)
                end
                finalScore = (student_score /
                    @assignment_information[:problem_average][problem]["total_done"].to_f * 100.0).to_i
                return finalScore.to_s + '%'
                #End Partial Credit
                #(@assignment_information[:problem_average][problem]["total_correct"].to_f /
                    #@assignment_information[:problem_average][problem]["total_done"].to_f
                    * 100.0).to_i.to_s + '%'
            end
        end
    end

    #Gets the student average for a problem in the student item report
    def get_student_average student
        if @assignment_information[:student_average][student.user.id].blank? or
            @assignment_information[:student_average][student.user.id]["total_done"] == 0
            return 'N/A'
        else
            #Partial credit calculations
            student_score = 0.0
            @assignment_information[:problem_list].each do |problem|
                student_score += partial_credit(student, problem)
            end
            return (student_score / @assignment_information[:student_average][student.user.id]["total_done"] *
                100.0).to_i.to_s + '%'
            #End partial credit

            #return (@assignment_information[:student_average][student.user.id]["total_correct"].to_f /
                #
                @assignment_information[:student_average][student.user.id]["total_done"] *
                100.0).to_i.to_s + '%'
        end
    end

    #Gets the total class average for an assignment
    def get_class_average

```

```

total_done = 0
total_correct = 0
@assignment_information[:problem_list].each do |problem|
  total_done += @assignment_information[:problem_average][problem]["total_done"]
  total_correct += @assignment_information[:problem_average][problem]["total_correct"]
end
return 'N/A' if total_done==0
#Partial Credit
class_score = 0.0
@student_class_students.each do |student|
  @assignment_information[:problem_list].each do |problem|
    class_score += partial_credit(student, problem)
  end
end
return " unless total_done!=0
(class_score / total_done).to_i.to_s + '%'
#(total_correct.to_f * 100 / total_done).to_i.to_s + '%"
end

#Gets whether or not a student got a problem right in the student item report
def get_student_correctness student, problem
  log = get_log student, problem
  if log.nil?
    "<img src='/images/report/not_started_small.png'"
  elsif log.end_time.nil?
    "<img src='/images/report/in_progress_small.png'"
  elsif log.problem.open_response?
    if log.correct.blank?
      "(Ungraded Essay)"
    else
      essay_grade_color = 'red'
      if log.correct >= 0.75
        essay_grade_color = 'green'
      elsif log.correct >= 0.5
        essay_grade_color = 'orange'
      end
      return "<span style='font-size:150%;color:#{essay_grade_color}'>" +
(log.correct*@student_class.get_full_mark_setting).to_i.to_s +
      "</span>" +
      "<br />" +
      "Graded Essay:"
    end
  elsif log.correct.to_i == 1
    "<img src='/images/report/correct_small.png'"
  #INSERT EXTRA CREDIT IMAGES HERE
  elsif log.correct.to_i == 0
    if @student_class.credit_setting
      if (log.hint_count + log.attempt_count <= @student_class.credit_attempts.to_i) and
log.bottom_hint < 1
        "<img src='/images/report/incorrect_green_small.png'"
      elsif log.bottom_hint < 1
        "<img src='/images/report/incorrect_small.png'"
      else
        "<img src='/images/report/incorrect_yelshadow_small.png'"
      end
    end
  else

```

```

        <img src='/images/report/incorrect_small.png'>
      end
    end
  end
end

```

#### App\Controlllers\Teacher\credit\_controller.rb

```

class Teacher::CreditController < Teacher::DefaultController
  require 'rubygems'
  begin
    require 'gs!'
    require 'rsruby'
    include GSL::Fit
  rescue LoadError
    # not installed
  end

  def save_credit_settings
    if params[:commit] == "Save Changes"
      Setting.create_or_update @current_user.id,"credit_setting", Scope::TEACHER, true, true
      Setting.create_or_update @current_user.id,"credit_attempts", Scope::TEACHER, true,
params[:maxAttempts]
    end
    redirect_to('/preferences')
  end

  def credit_settings
    @home_link_only = true
    @disabled = false #!Setting.get_enabled_setting_value(@current_user.id, "credit_setting", Scope::TEACHER)
    @attempts = Setting.get_enabled_setting_value(@current_user.id, "credit_attempts", Scope::TEACHER)
  end
end
end

```

#### App\Views\Teacher\Credit\credit\_settings.rhtml

```

<center>
  <h1>Partial Credit Settings and Information</h1>
</center>
<br><br>

```

This feature allows students to receive partial credit for problems that they do not answer correctly on the first attempt. The partial credit is automatically calculated by ASSISTments based on the number of attempts and/or hints used by the student along with the maximum number of attempts and/or hints set by the teacher (see below).

```

<div id='field' ">
  <% form_tag((url_for :action => :save_credit_settings)) do %>
<div>
  <div>
    <img src='/images/report/correct_small.png'>
    &nbsp;&nbsp;&nbsp;Answered correctly after the first attempt without using a hint.
  </div><br>
  <div>
    <img src='/images/report/incorrect_green_small.png'>
    &nbsp;&nbsp;&nbsp;Answered correctly after the first attempt and/or hint used. Received partial credit.
  </div><br>
  <div>
    <img src='/images/report/incorrect_small.png'>
    &nbsp;&nbsp;&nbsp;Answered correctly after exceeded the allotted number of attempts/hints. No partial credit.
  </div><br>

```



### App\Models\student\_class.rb

```
has_one :credit_setting

def credit_setting
  self.teachers.each do |teacher|
    return Setting.get_enabled_setting_value(teacher.user_id, 'credit_setting', Scope::TEACHER)
  end
end

def credit_attempts
  self.teachers.each do |teacher|
    return Setting.get_enabled_setting_value(teacher.user_id, 'credit_attempts', Scope::TEACHER)
  end
end
```

### App\Views\Teacher\\_submenu.rhtml

```
<% render_submenu(controller, self) do |m| %>
  <%= render_home_link %>
  <% if @home_link_only.nil? %>
  <span id="classes_menu"><a href="javascript:void(0);">Classes</a></span>
  <span id="reports_menu"><a href="javascript:void(0);">Reports</a></span>
  <% end %>
<% end %>
```

### Db\Migrate\329\_create\_credit\_settings.rb

```
class CreateCreditSettings < ActiveRecord::Migration
  def self.up
    ActiveRecord::Base.connection.execute("insert into default_settings (released_feature, in_use, invalidates, key_name,
    setting_value_format, setting_value, display_name, component, controller_path, settings_link_text, updated_at, description)
    values(true, true, false, 'credit_setting', 'Boolean', false, 'Partial Credit For Student Work', 'TEACHER',
    '', 'now()', '')")
    ActiveRecord::Base.connection.execute("update default_settings set display_position = id where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("insert into scope_settings (scope_id, default_setting_id, exposure) values
    ({Scope::TEACHER}, (select id from default_settings where key_name = 'credit_setting'), 'public');")

    ActiveRecord::Base.connection.execute("insert into default_settings (released_feature, in_use, invalidates, key_name,
    setting_value_format, setting_value, display_name, component, controller_path, settings_link_text, updated_at, description)
    values(true, true, false, 'credit_attempts', 'NUMBER - ##', 3, 'Partial Credit For Student Work',
    'TEACHER', '', 'now()', '')")
    ActiveRecord::Base.connection.execute("update default_settings set display_position = id where key_name
    = 'credit_attempts'")
    ActiveRecord::Base.connection.execute("insert into scope_settings (scope_id, default_setting_id, exposure) values
    ({Scope::TEACHER}, (select id from default_settings where key_name = 'credit_attempts'), 'public');")
  end

  def self.down
    ActiveRecord::Base.connection.execute("delete from scope_settings where default_setting_id in (select id from
    default_settings where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("delete from default_settings where key_name = 'credit_setting'")
    ActiveRecord::Base.connection.execute("delete from scope_settings where default_setting_id in (select id from
    default_settings where key_name = 'credit_attempts'")
    ActiveRecord::Base.connection.execute("delete from default_settings where key_name = 'credit_attempts'")
  end
end
```

## 9. References

---

- <sup>1</sup> <http://www.imsglobal.org/lti/ltiv1p2pd/ltiCIMv1p0pd.html> “IMS Global Learning Tools Interoperability™ Content-Item Message.” *IMS Global Learning Consortium*. Web.
- <sup>2</sup> <http://www.imsglobal.org/LTI/v1p1/ltiIMGv1p1.html> “IMS Global Learning Tools Interoperability™ Implementation Guide.” *IMS Global Learning Consortium*. Web.
- <sup>3</sup> <http://www.wpi.edu/Pubs/ETD/Available/etd-052914-115757/unrestricted/hduong.pdf> “Feature-Oriented Software Engineering Approach to Integrate ASSISTments with Learning Management Systems.” *Hien D. Duong*. Pdf.
- <sup>4</sup> [http://www.spvsoftwareproducts.com/java/lti\\_tool\\_provider/](http://www.spvsoftwareproducts.com/java/lti_tool_provider/) “Java LTI Tool Provider Package” *SPV*. Web.
- <sup>5</sup> <https://sites.google.com/site/assistmentsstestbed/lti> “LTI - How to Create Controlled Experiments in ASSISTments.” *ASSISTments Test Bed*. Web.
- <sup>6</sup> <https://github.com/instructure/ims-lti> “Instructure - IMS-LTI.” *GitHub*. Web.
- <sup>7</sup> <https://rubygems.org/gems/oauth> “Oauth 0.4.7.” *RubyGems*. Web.
- <sup>8</sup> [https://github.com/instructure/lti\\_tool\\_consumer\\_example](https://github.com/instructure/lti_tool_consumer_example) “Instructure-lti\_tool\_consumer\_example.” *GitHub*. Web.
- <sup>9</sup> <http://www.imsglobal.org/cc/statuschart.cfm> “IMS Interoperability Conformance Certification Status.” *IMS Global Learning Consortium*. Web.