

GDOT eLearning Customer Acceptance Form

This Customer Acceptance Form is used to obtain customer sign-off when a lesson, module or project deliverable is complete

Project:	
Deliverable:	
This document has been issued by:	Date Issued:

The Acceptance Criteria below specifies the deliverables, test criteria, quality standards, checkpoints, and payoffs agreed upon by the project team. The Acceptance Criteria serves two purposes: It acts as a primary quality control (QC) measurement, and as a contractual agreement between the training developer and customer, and defines success.

Acceptance Criteria

SME	Depth: Does the content provide sufficient depth of the subject areas, or is more information needed? Accuracy: Is the content accurate, correct and up-to-date? Specificity: Does the content need to be more specific in certain portions? Completeness: Does the content need to cover more areas, and are the subject areas discussed covered sufficiently? Does the content include all information necessary for the learner to master the stated learning objectives? Relevance: Does the content directly address informational needs? Is there extraneous information that can/should be eliminated, or is it necessary for understanding? Does the content include information that can/should be included in other lessons/materials?	
INSTRUCTIONAL DESIGN	Structure: Does the course have a clear beginning, middle and end? Are Lesson/course overviews provided, and is there a course/lesson review? Appropriateness: Is the complexity of the content sufficient for the intended audience? Soundness: Does the content have sound instructional design? Can content be better presented? Learning objectives: Have the learning objectives been clearly stated? Are the Learning Objectives measurable? Does the content support the learning objectives? Do the learning objectives themselves meet desired learning outcome for the project? Assessments: Are valid and reliable test assessment items included to measure mastery of the lesson/course learning objectives? Are the the quiz/test/assesment items sound?	
TECHNICAL	Style: Does the writing and format of the content adhere to recommended agency style conventions and is its usage consistent? Quality: Is the writing of good quality, coherent? Does the writing need minor editing? Are all images and graphics of sufficient high quality? Document Elements: Does the deliverable have all required content element such as images, graphics, buttons, audio, video, animation and text (including those for for captions, cutlines, callouts, popups?) Interactions: Are learning interactions provided (self checks, activities, games, show-me's, try-me's, and popups) Are the interactions meaningful, effective and support the objectives. Effectiveness: Do all graphics, images and other document elements support the content/text, and are they effective?	

GDOT eLearning Customer Acceptance Form

FINAL	Test Criteria: Users can access and use the WBT through the intranet or the Internet Standards: Time on task to complete a single lesson is 10 minutes or less. The product is user friendly and runs as expected. All links and buttons function correctly. All graphics and images load sufficiently. The deliverable runs as expected. Checkpoints: Progress checks to be conducted periodically and upon completion each lesson and module. Development of new lessons and modules will not commence until standards are met on current deliverable.	

The Project Outcome has been measured against its acceptance criteria and has been formally accepted on behalf of the customer. This project deliverable may now be closed.

Additional Comments about the Customers Acceptance:

Recorded Shortfalls of the Final Project Outcome (if any):

Subject Matter Expert:	Signature: _____ Name: _____ Date: _____
Instructional Designer	Signature: _____ Name: _____ Date: _____
Content developer	Signature: _____ Name: _____ Date: _____
Project Manager:	Signature: _____ Name: _____ Date: _____
Training Director	Signature: _____ Name: _____ Date: _____

Minimum Technical Requirements
Web-based training deployment on GDOT ELMs

(September 15, 2011)

1. Purpose

The purpose of this document is to identify the minimum requirements that must be fulfilled by content developers delivering course content to ELMs customers. This document also identifies the steps that customers normally take for reviewing, testing, piloting and rolling out course content on GDOT networks.

2. Audience

This document is meant for content developers who develop SCORM 1.2 compliant courses for delivery using ELMs. We encourage ELMs customers to point their content development teams and vendors to this document as a guideline. We also encourage all learning solution community members including content developers to edit this content or add new guidelines based on their experience.

2. General

- a. *Must be produced using a licensed product such as Lectora, Adobe Captivate or Adobe Flash. All software used in development must be reviewed and approved by GDOT's IT group.*
- b. *Content developers shall not use subscription-based authoring tools shall be used to develop product as this would necessitate that GDOT purchase a subscription to update the course.*
- c. *All vendors should ask for and obtain information about the Customer's standard hardware and software profiles*
- d. *Courseware packages must be compliant to SCORM 1.2 standards*
- e. *SCORM packages must include a imsmanifest.xml file at the root level*
- f. *Course should clearly indicate development/revision date, or version for version control*
- g. *Courseware content should be broken up into multiple SCOs if training is longer than 20 continuous minutes.*
- h. *Bookmarking, if employed, should be implemented in the course code at a minimum at the end of each course lesson, optimally at the page level.*
- i. *No server side scripts or executables such as .asp, .jsp, .cgi, or .exe files.*
- j. *Custom javascript functions or methods shall be submitted for code review prior to use within the course and shall be included in a separate .js file in the root of the course.*
- k. *GDOT will provide no Cold Fusion support .cfm, .cfml*

- l. No Streaming media or video. Clients who want to have streaming media in their custom content should make alternative arrangements for the physical location of such content.*
- m. Must provide all source files upon delivery of Web-based training product.*
- n. GDOT reserves the right to refuse file sets based on objectionable content, file types, file sizes, for example, files over 50MB per course or file set.*
- o. SCO's should be sized in the range of 1-40 MB. Customer has successfully run larger SCOs, though larger SCOs will require extra remote testing (field users on high speed connections) Courses may be made up of a single SCO as long as size issues do not affect remote playback. A large, single SCO course must display progress to the student.*
- p. If the size of any SCO or individual file affects remote viewing, adjustments must made to content so remote viewing (via high speed connection) is unencumbered by file sizes.*
- q. All SCOs in the product are expected to pass the ADL Test Suite 1.27 ST (see <http://www.adlnet.gov/>)*

3. Content requirements

- a. Shall include contact information and name of the course manager or SME.*
- b. Shall generally provide no links to external Web sites.*
- c. Shall provide clearly specified learning objectives.*
- d. Shall provide language for proper SCO exiting procedures. (Ex. "Do not use 'X' in the top right corner of the course window to exit the "lesson.")*
- e. Shall provide language at the end of assessments indicating whether the user has passed or failed the assessment.*
- f. Shall be compliant with Section 508 of the US Rehabilitation Act.*
- g. Shall satisfy user expectation of what is needed for successful completion*
- h. Shall satisfy user expectation regarding time commitment for course and each lesson.*
- i. Shall provide for information as to how far along the learner is in a lesson.*
- j. Shall identify copyrighted materials used in the courser either be citing that material appropriately or be obtaining written permission to use it online, in advance of the beginning of the course.*
- k. Shall minimize horizontal and vertical scrolling*

4. Other Requirements:

- a. Content developers shall provide narrative descriptions of the course, its modules and its lessons. This is information is required in order to set up the course in ELMS.*
- b. A course description, called a "Catalog Abstract" shall not be more than 4000 characters with spaces.*

- c. *A module description, called a “Component Description” is limited to 254 characters with spaces.*
- d. *A lesson description, called an “Item Description” is also limited to 254 characters with spaces.*

5. GDOT ELMS Standards Compliance

- a. *GDOT ELMS currently supports the following standards:*

- *SCORM 1.2*
- *AICC-HACP*

- b. *SCORM 1.2 file requirements*

For SCORM 1.2, compliance, content developers must provide, in addition to the course files and folders, the following course structure files at the root level:

- i. *Adlcp_rootv1p2.xsd*
- ii. *Ims_xml.xsd*
- iii. *Imscp_rootv1p1p2.xsd*
- iv. *Imsmanifest.xml*
- v. *Imsmd_rootv1p2p1.xsd*
- vi. *Standard.js*
- vii. *In addition, all SCORM 1.2 compliant courseware shall contain an IMS Manifest zip file:*
- viii. *Imsmanifest.xml*

- c. *AICC-HACP file requirements*

For AICC-HACP compliance, content developers must provide, in addition to the course files and folders, the following course Structure files at the root level:

- i. **.au*
- ii. **.crs*
- iii. **.des*
- iv. **.cst*
- v. *Standard.js*

6. Supported SCORM 1.2 Methods

- a. *Must identify how each SCO is to communicate with ELMS (which methods are called and data elements initialized) SCORM 1.2 specifies eight methods:*
 - i. *LMSInitialize (Initializes the SCORM API)*

- ii. *LMSGetValue (Gets the value of a parameter [student_id, student_name, score, total time, lesson_Status, lesson_location, objectives score])*
- iii. *LMSSetValue (Sets the value of a parameter [score,session_time, lesson_status, objectives score])*
- iv. *LMSCommit (Instructs the LMS to persist the new values in the database)*
- v. *LMSFinish (Indicates end of lesson)*
- vi. *LMSGetLastError (Gets the error code)*
- vii. *LMSGetErrorString (Get the error string for the error code)*
- viii. *LMSGetDiagnostic()*

b. *LMSInitialize()*

- i. *Must initialize the SCO on load of the very first page (before any actual content is loaded),*
- ii. *Must be called by the onload event handler in the <body> tag of a course's first HTML page.*
- iii. *This first page may include course content or may be a container that holds the actual course content in an embedded Flash movie or inside another HTML frame.*
- iv. *JavaScript Sample:*

```
theAPI.LMSInitialize()
```

c. *LMSSetValue()*

- i. *Must submit any data to ELMS as soon as it's available*
- ii. *The data element identified in the parameter of LMSSetValue() must adhere to rules for that data element regarding data type, range, and dependency;*
- iii. *The data element identified in the parameter must not be defined as "read-only."*
- iv. *JavaScript Sample:*

```
theAPI.LMSSetValue("cmi.core.lesson_location",  
mylocation.toString() + "")
```

d. *LMSGetValue()*

- i. *The LMSGetValue() method provides the means for a SCO to retrieve the value assigned to a SCORM Data Element*
- ii. *The data element identified in the parameter must have been initialized with a value (either by the ELMs or by the SCO)*
- iii. *The data element identified must not be defined as "write-only."*
- iv. *JavaScript Sample:*

```
mylocation=theAPI.LMSGetValue("cmi.core.lesson_location")
```

e. *LMSCommit()*

- i. *Must submit any data to ELMS as soon as it's available*
- ii. *Shall not be called with each page turn nor be excessive to slow the course.*
- iii. *Must be called at regular intervals or key instances so as to prevent any potential loss of learner data resulting from, for example power failure, system crashes, network errors, etc... LMSCommit intervals shall occur once every 10-15 minutes, or at the completion of a module/lesson point, providing that the lesson point is every 10-15 minutes, or at the onunload event.*
- iv. *Must be called along with the LMSSetValue method whenever the user exits the course*
- v. *JavaScript Sample:*

```
theAPI.LMSCommit()
```

f. *LMSFinish()*

- i. *Must terminate the SCO on unloading the last page.*
- ii. *Must be dispatched in either of the following circumstances:*
 - 1. *The user clicks a designated "Exit" button within the course, a "soft close".*

2. *The user ends the course by means of a "hard close". A hard close typically occurs when the browser window that contains course content is closed. In the case of Windows/Internet Explorer, a hard close occurs when the user clicks the red 'x' in the top right of the window.*
- iii. *Must be called in the <body> tag of a course's first HTML page. The onunload event handler should trigger LMSFinish(). This is done to provide a single trigger for exiting the course whether the user leaves using a designated exit button within the course or if they initiate a "hard close" by clicking the windows 'x' icon.*
- iv. *JavaScript Sample:*

```
theAPI.LMSFinish()
```

7. Supported SCORM 1.2 Data Elements

- i. *Must specify required data elements used for the methods LMSGetValue() and LMSSetValue().*
- ii. *ELMs data elements for LMSGetValue() are:*
 1. *cmi.core.student_id*
 2. *cmi.core.student_name*
 3. *cmi.core.credit*
 4. *cmi.core.lesson_status*
 5. *cmi.core.entry*
 6. *cmi.core.total_time*
- iii. *ELMs data elements for LMSSetValue() are:*
 1. *cmi.core.lesson_status*
 2. *cmi.student_data.mastery_score*
 3. *cmi.core.score.raw*

8. Bookmarking

- a. *Delivered SCO's should be in a size/format that allows bookmarking and progress measurement. If there are multiple lessons within a SCO, internal course programming must perform book-marking.*
- b. *The cmi.core.lesson_location data element , when configured, allows a SCO to store a "bookmark" in order to allow a student to return to a point in a SCO where they previously left off.*

- c. If bookmarks are employed, this data element shall be set and committed when the learner passes course milestones, or every time the learner navigates forward 5 screens.
- d. While the lesson_location data element is a string, it must be determined by the SCO and should consist of a single integer to refer to a screen number within the course. ELMS does not interpret this value, but simply makes it available to the SCO upon a user's return to that SCO.
- e. The javascript example below shows an example of how an elearning course uses the LMSSetValue() function to set a bookmark in an elearning course.

```
theAPI.LMSSetValue("cmi.core.lesson_location", mylocation.toString() + "")
```

- f. To make sure that the learning progress is set correctly, the content should set cmi.core.lesson_status value explicitly by each SCO. Each SCO needs to set the cmi.core.lesson_status to "completed" for the overall course progress to be marked complete.

9. Optional Data Element support

- a. Depending on a content developer's business requirements, GDOT may need to configure certain optional data elements so the SCO can initialize and report the these optional elements .
- b. Content developers must specify in advance which data elements are needed. For example, if your business requirements dictate that data elements for any cmi.interactions are needed, these must be specified in advance. These include:
 - i. the time spent on the interaction,
 - ii. what the interaction type was (a multiple-choice question, for example),
 - iii. what the question text was,
 - iv. the answers, and which answer the learner chose,

Except for finding and communication with the API adaptor object, a SCO is not allowed to communicate or manipulate other windows that exist in the environment provided by ELMS.

A SCO may open a dependent popup window but this not considered a best practice. In any case, the SCO is responsible for closing such a window before it is unloaded. a SCO is not allowed to leave any trace of itself in the user's environment after it has been unloaded.

Supported SCORM 1.2 Data Elements (cont'd)

The program may include only mandatory SCORM elements and the optional SCORM elements supported by ELMS (see appendix 1). At minimum, the content must include the following SCORM elements:

GDOT ELMs (known) supported data elements highlighted:

	Get	Set	Type	Description
CMI (top-level elements)				
cmi._version	Yes	No	CMIStr255	(as per SCORM spec)
cmi._children				
cmi.suspend_data	Yes	Yes	CMIStr4096	Start with empty string. Must be a string.
cmi.launch_data	Yes	No	CMIStr4096	Initialization Parameters for the SCO. Always the same. Comes from MANIFEST. Default to "" (empty string)
CMI.CORE				
cmi.core._children				
cmi.core.student_id	Yes		CMIIIdentifier	username
cmi.core.student_name	Yes		CMIStr255	Lastname, firstname
cmi.core.lesson_location				
cmi.core.credit	Yes			credit
cmi.core.lesson_status	Yes		CMIVocabulary	passed, completed, failed, incomplete, browsed, not attempted
cmi.core.entry	Yes		CMIVocabulary	ab-initio, resume, "" (empty string)

<i>cmi.core.score._children</i>				
<i>cmi.core.score.raw</i>		Yes	CMIDecimal/ CMIBlank	a normalized value between 0 and 100, initially blank
<i>cmi.core.score.min</i>				
<i>cmi.core.score.max</i>				
<i>cmi.core.total_time</i>	Yes		CMITimespan	Initially is "00:00:00.00" – total time for all sessions
<i>cmi.core.exit</i>				
CMI.STUDENT_DATA				
<i>cmi.student_data._children</i>				
<i>cmi.student_data.mastery_score</i>	Yes	Yes	CMIDecimal	A normalized value between 0 and 100, comes from the MANIFEST – either there or not.

Content Guidelines for Developers

5. Basic Requirements

5.1. High-Level Technical Requirements

5.1.1. GDOT currently uses ELMS 9.0 Learning Management System.

5.1.2. Filenames are case sensitive so all programming calls for files must match the case of the file name, for example, if the program calls file "Custom.xml", "Custom.xml" must exist in the course directory, not "custom.xml."

5.1.3. The vendor must deliver content in SCORM 1.2 format with an imsmanifest.xml file included to import/publish the content.

5.1.4. The program may include only mandatory SCORM elements and the optional SCORM elements supported by ELMS (see appendix 1). At minimum, the content must include the following SCORM elements:

~~5.1.5. To make sure that the learning progress is set correctly, the content should set cmi.core.lesson_status value explicitly by each SCO. Each SCO needs to set the cmi.core.lesson_status to "completed" for the overall course progress to be marked complete.~~

~~5.1.6. All SCOs in the product are expected to pass the ADL Test Suite 1.27 ST (see <http://www.adlnet.gov/>)~~

~~5.1.7. Delivered SCO's should be in a size/format that allows bookmarking and progress measurement. If there are multiple lessons within a SCO, internal course programming must perform book marking.~~

~~5.1.8. SCO's should be sized in the range of 1-40 MB. Customer has successfully run larger SCOs, though larger SCOs will require extra remote testing (field users on high speed connections). Courses may be made up of a single SCO as long as size issues do not affect remote playback. A large, single SCO course must display progress to the student.~~

~~5.1.9. If the size of any SCO or individual file affects remote viewing, adjustments must made to content so remote viewing (via high speed connection) is unencumbered by file sizes.~~

5.1.11. Self-tests may be offered throughout the program. The results of such tests will not be stored though a pass/fail indicator can be set using SCORM and programming commands. This status can be used as a prerequisite for accessing later SCOs.

5.1.12. End-of-course assessments can be written in any format. Only SCORM 1.2 parameters (pass/fail and score) parameters can be passed to ELMS. Data regarding the tests can be stored in the Suspend Data field in the ELMS SCORM result tables. One set of Suspend Data can be stored per SCO.

5.1.13. The course assets will be stored in the content folder on ELMS. GDOT must approve links to sources/content outside the firewall.

5.1.14. Content will not automatically launch full screen. If the content is to run in the content player, the sizing must address the content player borders including any headers and footer height considerations if applicable.

6. Tools & Technologies

6.1. Development Tools

6.1.1. The media types and technologies used in the program must, wherever possible, be based on open standards and/or compliant with commonly accepted industry standards and best practices.

~~6.1.2. All software used in development must be reviewed and approved by Customer's IT group.~~

~~6.1.3. All vendors should ask for and obtain information about the Customer's standard hardware and software profiles.~~

7. Reviews, Tests, and Pilots

Internal Testing: All products intended to be launched on ELMS will be reviewed, tested and piloted internally prior to course rollout.

7.1. Content Review

7.1.1. It is preferable that web-based materials for content review are provided in a simple web-based format, e.g., launched from a static web link so that content reviewers can review product without the completion requirements enforced by a SCORM format. In other words, reviewers should be able to navigate at will, so if they want to review content on page six of lesson three, they don't have to complete lessons one and two as well as listen/read all of the content of pages 1-7 in lesson 3. If delivered in SCORM format, vendors should instruct Customer on how to run the test in debugger mode so that reviewers can navigate at will.

7.1.2. Navigation instructions within, between SCO's must be approved by GDOT

7.2. Navigation Testing

7.2.1. If the content is in a new delivery format, a test SCO (or two) must be sent during the development process to make sure that GDOT approves of the format and navigation.

7.2.2. GDOT may perform tests on partially completed content as long as it is in the format that will be used for the final product.

7.2.3. When the web content (SCORM version) is completed for any part of the project, GDOT will publish the course in the ELMS test environment.

7.2.4. GDOT will complete initial testing of navigation, pop-ups, process indications, bookmarking, completion records, etc. The content developer is welcome to participate in this process upon the approval by GDOT.

7.2.5. GDOT will require a full course review with the delivery and publishing of any new content or file change.

7.3. Piloting Complete Materials

7.3.1. Pilots can be accommodated throughout the development process. GDOT can provide web links for content review as well as published materials (via ELMS Training environment for navigation/testing review).

Appendix 1: Optional SCORM Elements supported by SAP's Learning Solution 2.00

SAP Learning Solution is an LMS LMS-RTE2 and supports mandatory elements and some optional elements.

Element	Mandatory	Supported	Remark
cmi.version	yes	yes	
cmi.core.student_name	yes	yes	
cmi.core.student.id	yes	yes	
cmi.core.lesson_location	yes	yes	
cmi.core.credit	yes	yes	Only field value „credit“ supported
cmi.core.lesson_status	yes	yes	
cmi.core.entry	yes	yes	
cmi.core.score.raw	no	yes	
cmi.core.score.min	no	yes	
cmi.core.score.max	no	yes	
cmi.core.total_time	yes	yes	
cmi.core.lesson_mode	no	yes	Only field value „normal“ supported
cmi.core.exit	yes	yes	
cmi.core.session_time	yes	yes	
cmi.suspend_data	yes	yes	
cmi.launch_data	yes	yes	
cmi.comments	no	yes	
cmi.comments_from_lms	no	no	
cmi.objectives	no	no	All subelements are also not supported.
cmi.student_data	no	no	All subelements are also not supported.
cmi.student_preference	no	no	All subelements are also not supported.
cmi.interactions	no	no	All subelements are also not supported.

- External content must fulfill the following criteria to be interoperable with ELMS:
- All SCOs are of the type SCO-RTE1, at a minimum.
- They must actively support cmi.core.lesson_status of the mandatory elements.

This is particularly true for SCOs that implement (self-) tests. The content should integrate seamlessly in the user interface, and additional browser windows should only be opened for further information, such as Help or Glossary.

Labels parameters

Labels

[scorm](#)

[content](#)

[elearning](#)

[lms](#)

[lso](#)