

SAP SuccessFactors

INTERNAL – Authorized for SAP Customers and Partners
Document Version: 2H 2025 – 2026-04-13

Implementing Structured Content

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

SAP

Content

1 AICC and SCORM Support for SAP SuccessFactors Learning. 6

2 Deployment of Structured Content. 7

3 SCORM Overview. 8

3.1 SCORM Manifest. 8

3.2 SCORM Supported Data Elements. 12

3.3 SCORM 2004 API Calls Supported by SAP SuccessFactors Learning. 13

3.4 SCORM 2004 Sequencing. 22

Common SCORM Sequencing Terms. 22

Additional Resources for SCORM Sequencing. 23

SCORM 2004 Sequencing Root Elements. 23

`sequencing`. 24

`controlMode`. 24

`constrainedChoiceConsiderations`. 25

`sequencingRules`. 26

`preConditionRule`. 27

`postConditionRule`. 27

`exitConditionRule`. 28

`ruleConditions`. 28

`ruleCondition`. 29

`ruleAction`. 30

`limitConditions`. 31

`rollupRules`. 32

`rollupRule`. 33

`rollupConditions`. 34

`rollupCondition`. 35

`rollupAction`. 36

`rollupConsiderations`. 37

`objectives`. 38

`primaryObjective`. 39

`objective`. 40

`minNormalizedMeasure`. 40

`mapInfo`. 41

`randomizationControls`. 42

`deliveryControls`. 43

`presentation`. 44

navigationInterface . . . 45
hideLMSUI . . . 45
SAP SuccessFactors Learning Navigation Controls for SCORM 2004 Content . . . 47
3.5 SCORM API . . . 47
3.6 SCORM 1.2 . . . 49
SCORM 1.2 API Calls Supported by SAP SuccessFactors Learning . . . 49

4 AICC Overview . . . 54
4.1 The Difference Between AICC Appendix A and Appendix B . . . 55
4.2 SAP SuccessFactors Learning AICC Support . . . 55
4.3 AICC Data Format . . . 55
4.4 Standard AICC_DATA Values . . . 56
AICC [Core] Group Overview . . . 58
AICC [Core_Vendor] . . . 62
AICC [Core_Lesson] . . . 63
Example of AICC Supported Data Elements . . . 63
4.5 AICC Max_Normal . . . 64
Setting AICC Max Normal Default . . . 64
Setting AICC Max Normal In a Course . . . 65
AICC Max_Normal Inheritance . . . 66
4.6 Supported AICC Import and Export Files . . . 66
4.7 When and How to Use the AICC Document Wrapper . . . 67
Using the External AICC Wrapper to Create Confirmation Statements . . . 68
Creating Content That Users Can Confirm Reading . . . 69
Changing Internal AICC Wrapper Text Confirmation Text . . . 70
Setting AICC Wrapper Confirmation Text . . . 71

5 Structured Content Troubleshooting . . . 73
5.1 Checking the Launch Method of Online Content . . . 73
5.2 Checking to See if Online Content is Communicating Status About a User . . . 74
5.3 Using Fiddler to Watch Traffic Between Course Content and SAP SuccessFactors Learning . . . 75
5.4 Communication Between SCORM 1.2 Learning Content and SAP SuccessFactors Learning . . . 75
5.5 Common Problems and Solutions for SCORM 1.2 and AICC Content . . . 76
5.6 Content Object Launch Method Fields in SAP SuccessFactors Learning . . . 78
5.7 Potential Integration Issues . . . 82
5.8 Structured Content APIs Depend on Proper Usage of Session . . . 83
5.9 SCORM 1.2 to SCORM 2004: Sequencing . . . 84
SCORM 1.2 to 2004 Run Time Environment (RTE) Changes . . . 84
SCORM 1.2 to 2004 Run Time Environment (RTE) Function Name Changes . . . 85
SCORM 1.2 to 2004 Run Time Environment Additions . . . 85
SCORM 1.2 to 2004 Core Data Group Changes . . . 87
SCORM 1.2 to 2004 Content Aggregation Model Changes . . . 88

5.10 Characters Not Supported in Content File Names. 88

6 Cross Domain Solution Troubleshooting. 90

6.1 Using Test Pages to Troubleshoot Communication. 91

6.2 Cross Domain Solution at Run Time. 91

7 Configuring Cross Domain Solutions. 93

7.1 Getting the Latest Proxlet WAR for Cross-Domain Content (CDC). 93

7.2 Setting Up the Cross Domain Solution on a Content Server. 94

Technical Requirements for the Cross Domain Solution (CDS). 95

Proxlet Cross Domain Solution Overview. 95

7.3 Configuring LMS_ADMIN for the Cross Domain Solution. 97

Communication Cross Domain Support Settings. 98

7.4 Enabling the Cross Domain Solution on Content Objects. 99

Change History

Learn about changes to the documentation for `<Delivering Structured Content>` in recent releases.

1H 2021

Type of Change	Description	More Info
None	We did not update this document.	

2H 2020

Type of Change	Description	More Info
None	We did not update this document.	

1 AICC and SCORM Support for SAP SuccessFactors Learning

SAP SuccessFactors Learning supports these versions of structured online leaning content.

- AICC
- SCORM 1.2
- SCORM 2004 2nd Edition
- SCORM 2004 4th Edition

2 Deployment of Structured Content

Although we recommend that customers use Premium Content Management (PCM), some customers self-manage their deployments of structured content.

If you want to self-manage your content or you want to outsource the management to a third-party vendor, we offer support in the form of troubleshooting and supported structure documentation. We recommend that if you self-manage your content or you use a third-party vendor, that you familiarize yourself with the supported structures, API calls, and data elements for SAP SuccessFactors Learning.

We support three types of self-managed deployments of structured content:

- AICC
- SCORM (1.2, 2004 2nd Edition, and 2004 4th Edition)
- Native Tracking Functions

3 SCORM Overview

The Sharable Content Object Reference Model (SCORM) is a model that references a set of interrelated technical specifications and guidelines designed to meet the Department of Defense's high-level requirements for Web-based learning content.

SCORM was developed by the Advanced Distributed Learning (ADL) initiative for the Department of Defense as a means of constructing and presenting learning objects. It is a combination of several different pre-existing standards. SCORM treats each individually referenced specification as a separate book. SCORM 1.2 and SCORM 2004 have a different configuration of these books.

3.1 SCORM Manifest

The SCORM manifest is SCORM files used by the SAP SuccessFactors Learning for importing course definitions.

SCORM content usually contains files which are used by a Learning Management System (LMS) to import the course definitions. The primary import file is the `imsmanifest.xml`, which may or may not have supporting XML files. The actual content is **not** imported into the LMS, just the definition or setup of the content (the Content Package and Content Object definitions).

A SCORM `imsmanifest.xml` file can be divided into three primary parts.

- Resources
- Organizations
- Supporting metadata

Resources Section

The resources section is composed of a single `resources` element, which contains one or `resource` elements. Individual <resource> elements define individual pieces of content which may be either an Asset or a SCO. An asset is simply a piece of content which does not contain code to communicate SCORM data. A SCO is content which contains code for SCORM communication.

Attribute of `resource` element	Description
`adlcp:scormtype`	Identifies the resource as either an "asset" or a "sco"
`identifier`	Unique ID for the resource. Used in the Organizations section to reference the resource.
`type`	defines the type of resource. Always defined as "webcontent".

Attribute of `resource` element	Description
`href`	Required for SCO resources. This is the filepath to the resource.

Sample Code

```
<resources>
            <resource identifier="R_S100004" type="webcontent"
                   adlcp:scormtype="sco" href="Course01/Lesson01/sco01.htm">
          </resource>
          <resource identifier="R_D1" adlcp:scormtype="asset"
                 type="webcontent" xml:base="Course01">
            <file href="SCOFunctions.js"/>
            <file href="APIWrapper.js"/>
          </resource>
       </resources>
```

Organizations

The Organizations section is composed of a single `organizations` element which typically contains a single, but may contain multiple, `organization` elements. An organization element defines a course structure using one or more `item` elements. An item may map to a `resource` resulting in a content object, or an item may contain other item elements resulting in a folder in the course structure. In SAP SuccessFactors Learning, an item referencing a "sco" type resource is imported as a SCORM type content object and an item referencing an "asset" type resource is imported as a content object that is configured to "mark complete on launch".

Attributes of *item* Element	Description
`identifierref`	Ties the item to a resource through the resource identifier attribute.
`identifier`	Unique ID for the item.
`parameters`	Used to define URL parameters to be passed to the content.
`isvisible`	Indicates if the item is visible in the LMS course structure.

Sample Code

```
<organizations default="B0">
            <organization identifier="B0"
             <title>Maritime Navigation</title>
            <item identifier="B110" isvisible="true">
               <title>Steering &amp;: Sailing Rules</title>

                <item identifier="S110001" identifierref="R_S110001">
                <title>Conduct of Vessels in any Condition of Visibility</title>
               </item>
               <item identifier="S110002" identifierref="R_S110002">
                <title>Conduct of Vessels in Sight of One Another</title>
               </item>
            </item>
          </organization>
```

```
        </organizations>
```

The above sample would result in the following structure:

- Steering & Sailing Rules
  - Conduct of Vessels in any Condition of Visibility
  - Conduct of Vessels in Sight of One Another

Metadata Section

A `metadata` element can contain a wide array of information most of which is irrelevant to an LMS. SAP SuccessFactors Learning pulls only the following elements from supporting `metadata`.

- Title - Defined in a `title` element under the `general` element
- Description information - defined in a `description` element under the `general` element.
- Educational Objective - The objective is pulled from the `description` element under a `classification` element but only if the `puporse` element for the `classification` contains a `value` element defined as "Educational Objective".

Sample Code

```
<general>
          <title>
           <langstring>Conduct of Vessels in any Condition of Visibility</
langstring>
        </title>
        <description>
           <langstring>
               Discusses general rules of operation for vessels on inland
               waters. Topics discussed include: Look-out, Safe Speed, Collision,
               Channels, Traffic Separation.
           <langstring>
         </description>
       </general>

       <classification>
        <purpose>
           <source>
              <langstring xml:lang="x-none">LOMv1.0</langstring>
           </source>
           <value>
              <langstring xml:lang="x-none">Educational Objective</langstring>
           </value>
        </purpose>
        <description>
           <langstring>
               This Sharable Content Object will give the student a basic
               understanding of the conduct of vessels in any condition of
               visibility.
           <langstring>
         </description>
       </classification>
```

SAP SuccessFactors Learning Content Package (Course-level)

LMS Field	Manifest Document	Section	Data Element	Condition
Title	imsmanifest.xml	Organizations	Organization.title	
Description	Organization metadata	General	Description	
Objective	Organization metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose"
Folder label	imsmanifest.xml	Organization	Item.title	If Item data element contains sub items
Content object label	imsmanifest.xml	Organization	Item.title	If Item identifierref attribute maps to an asset or SCO resource

SAP SuccessFactors Learning Content Objects (SCO-level)

LMS Field	Manifest Document	Section	Data Element	Condition
ID	imsmanifest.xml	Resources	Resource.identifier	auto-generates content object ID from the first 15 characters of the manifest identifier, an underscore and then the first 12 characters of the resource identifier. <first 15 of manifest identifier>_<first 12 of resource identifier>
Title	resource metadata	General	Title	
Description	resource metadata	General	Description	
Launch path	imsmanifest.xml	Resources	href	If Resource is referenced in the <organization>
Objective	resource metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose". The metadata for Purpose>Source>langstring should be "LOMv1.0".

3.2 SCORM Supported Data Elements

View this section if you want to know about SCORM supported data elements.

SCORM Supported CAM Data:

SAP SuccessFactors Administration Content Package (Course-level)

SAP SuccessFactors Administration Field	Manifest Document	Section	Data Element	Condition
Title	imsmanifest.xml	Organizations	Organization.title	
Description	Organization metadata	General	Description	
Objective	Organization metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose"
Folder label	imsmanifest.xml	Organization	Item.title	If Item data element contains sub Items
Content object label	imsmanifest.xml	Organization	Item.title	If Item identifierref attribute maps to an asset or SCO resource.

SAP SuccessFactors Administration Content Objects (SCO-level)

SAP SuccessFactors Administration Field	Manifest Document	Section	Data Element	Condition
ID	imsmanifest.xml	Resources	Resource.identifier	auto-generates content object ID from the first 15 characters of the manifest identifier, an underscore and then the first 12 characters of the resource identifier.
Title	resource metadata	General	Title	
Description	resource metadata	General	Description	
Launch path	imsmanifest.xml	Resources	href	If Resource is referenced in the <organization>
Launch parameters	Imsmanifest.xml	Item	parameters	Additional url parameters for a given SCO

SAP SuccessFactors Administration Field	Manifest Document	Section	Data Element	Condition
Objective	resource metadata	Classification	Description	Only Classification sections with "Educational Objective" specified under "Purpose". The metadata for Purpose>Source>langstring should be "LOMv1.0".

3.3 SCORM 2004 API Calls Supported by SAP SuccessFactors Learning

SAP SuccessFactors Learning supports a set of SCORM 2004 API calls.

Call	Description
`Initialize( )`	The content must call this function before calling any other API function. It indicates to the LMS system that the content is going to communicate. The LMS can take any initialization steps required in this function.

Call	Description
`GetValue( )`	This function is used to pass data from the LMS to the content. Only one value is returned for each get request. The category and/or element is named in the parameter. The following `GetValue` parameters are supported: • `cmi._version` - Represents the version of the data model - LMS returns "1.0" • `cmi.learner_id` - The learner ID • `cmi.learner__name` - The learner name • `cmi.location` - 1000 char string usually used for bookmarking • `cmi.credit` - Indicates if SCO is taken for credit - returns "credit" or "no_credit" • `cmi.completion_status` - Indicates if the learner has completed the SCO - returns "completed", "incomplete", "not attempted", or "unknown" • `cmi.entry` - Indicates is the learner has previously launched the SCO - returns "ab_initio", "resume", or "" • `cmi.progress_measure` - Progress indicator - returns a value from 0.0 to 1.0 • `cmi.completion_threshold` - Value to determine SCO completion - Set in the LMS during import - returns a value from 0.0 to 1.0 • `cmi.success_status` - Indicates the learner has mastered the SCO - returns "passed", "failed", or "unknown" • `cmi.score._children` - Returns "scaled", "raw", "min", and "max" • `cmi.score.raw` - The raw score value • `cmi.score.max` - The maximum possible score - if not set, assumed 100 • `cmi.score.min` - The minimum possible score - if not set, assumed 0 • `cmi.score.scaled` - The score value scaled from -1.0 to 1.0 • `cmi.scaled_passing_score` - Value to determine SCO mastery - Set in the LMS during import - Returns a value from 0.0 to 1.0 • `cmi.total_time` - Returns the sum of all a learner's session times • `cmi.mode` - Indicates the SCO mode - Returns "browse", "normal", or "review"

Call	Description
	• `cmi.max_time_allowed` - Number of seconds a learner is allowed in a SCO - set in the LMS during import • `cmi.time_limit_action` - What the SCO should do when cmi.max_time_allowed is exceeded - set in the LMS during import - returns ""exit_message", "continue_message", "exit_no_message", or "continue_no_message" • `cmi.suspend_data` - 4000 char string usually used for progress data • `cmi.launch_data` - 4000 char string used to pass data to the SCO on launch from the LMS • `cmi.comments_from_learner.n.comment` - 4000 char string used to store learner comments where "n" is the comment number • `cmi.comments_from_learner._children` - Returns "comment", "location", "timestamp" • `cmi.comments_from_learner._count` - Returns the number of comments currently stored for the learner for this SCO. • `cmi.comments_from_lms.n.location` - 250 char string used to identify the location in the SCO to which the lms comment applies • `cmi.comments_from_learner.n.timestamp` - date and time comment created • `cmi.objectives._children` - Returns "id", "score", "success_status", "completion_status", and "description" • `cmi.objectives._count` - Returns the number of objectives currently stored for this SCO. • `cmi.objectives.n.id` - 4000 char string - returns the objective identifier where "n" is the objective number • `cmi.objectives.n.score_children` - Returns "scaled", "raw", "min", "max" • `cmi.objectives.n.score.scaled` - The objective score value scaled from -1.0 to 1.0 • `cmi.objectives.n.score.raw` - The raw objective score value • `cmi.objectives.n.score.max` - The maximum possible objective score - if not set, assumed 100 • `cmi.objectives.n.score.min` - The minimum possible score - if not set, assumed 0 • `cmi.objectives.n.progress_measure` - Progress indicator - returns a value from 0.0 to 1.0

Call	Description
	• `cmi.objectives.n.success_status` - Indicates the learner has mastered the objective - returns "passed", "failed", or "unknown" • `cmi.objectives.n.completion_status` - Indicates if the learner has completed the objective - returns "completed", "incomplete", "not attempted", or "unknown" • `cmi.objectives.n.description` - 250 char string objective description • `cmi.learner_preference._children` - Returns "audio_level", "language", "delivery_speed", "audio_captioning" • `cmi.learner_preference.audio_level` - Identifies learner preference for audio volume - returns number > or = 0 • `cmi.learner_preference.language` - 250 char string identifying the learner language preference • `cmi.learner_preference.audio_captioning` - Enables captions - returns "off", "no_change", or "on" • `cmi.learner_preference.deliverry_speed` - Identifies learner preference for presentation speed - returns number > or = 0 • `cmi.interactions._children` - Returns "id", "type", "objectives", "timestamp", "correct_responses", "weighting", "learner_response", "result", "latency", and "description" • `cmi.interactions._count` - Returns the number of interactions currently stored for this SCO • `cmi.interactions.n.id` - 4000 char string - returns the interaction identifier where "n" is the interaction number • `cmi.interactions.n.type` - Returns "true-false", "choice", "fill-in", "long-fill-in", "likert", "matching", "performance", "sequencing", "numeric", or "other" • `cmi.interactions.n.weighting` - Weight of the interaction for use in grading by the SCO • `cmi.interactions.n.learner_response` - Learner response - vocabulary depends on interaction type • `cmi.interactions.n.result` - Returns "correct", "incorrect", "unanticipated", "neutral", or a numeric estimate of correctness

Call	Description
	• `cmi.interactions.n.latency` - Time (seconds) spent answering the interaction • `cmi.interactions.n.description` - 250 char string description of the interaction • `cmi.interfactions.n.timestamp` - Time the interaction was presented to the learner • `cmi.interactions.n.objectives._count` - Returns the number of objectives currently stored for this interaction • `cmi.interactions.n.correct_response.n.pattern` - Returns a correct response pattern (answer) for the interaction - a single interaction may have multiple correct response patterns • `adl.nav.request` - Indicates the navigation request stored with the LMS - returns "continue", "previous", "choice", "exit", "exitAll", "abandon", "abandonAll", or "_none_" • `adl.nav.request_valid.continue` - Indicates if a "continue" navigation request is valid for this SCO - returns "true", "false", or "unknown" • `adl.nav.request_valid.previous` - Indicates if a "previous" navigation request is valid for this SCO - returns "true", "false", or "unknown" • `adl.nav.request_valid.choice.{target=STRING}` - Indicates if a "choice" navigation request is valid for this SCO where "target=STRING" is the SCO/module identified for navigation - returns "true", "false", or "unknown" The following are legitimate `GetValue` parameters, but are write-only. These would therefore return "" and set an error: 405 - element is write-only. • `cmi.core.exit` • `cmi.core.session_time`

Call	Description
`SetValue( )`	This function is used to pass data from the content to the LMS. The parameter indicates which category or element is being set. Only one value may be set with a single function call. The following SetValue parameters are supported: • `cmi.location` - 1000 char string usually used for bookmarking • `cmi.completion_status` - Indicates if the learner has completed the SCO - accepts "completed", "incomplete", "not attempted", or "unknown" • `cmi.exit` - Indicates how the learner left the SCO - accepts "time-out", "suspend", "logout", "normal", or "" • `cmi.progress_measure` - Progress indicator - accepts a value from 0.0 to 1.0 • `cmi.success_status` - Indicates the learner has mastered the SCO - accepts "passed", "failed", or "unknown" • `cmi.score.raw` - The raw score value • `cmi.score.max` - The maximum possible score - if not set, assumed 100 • `cmi.score.min` - The minimum possible score - if not set, assumed 0 • `cmi.score.scaled` - The score value scaled from -1.0 to 1.0 • `cmi.session_time` - Time spent in the SCO - PT(hours)H(min)M(sec)S • `cmi.suspend_data` - 4000 char string usually used for progress data • `cmi.comments_from_learner.n.comment` - 4000 char string used to store learner comments where "n" is the comment number • `cmi.comments_from_learner.n.location` - 250 char string used by the SCO to identify the location in the SCO to which the learner comment applies • `cmi.comments_from_learner.n.timestamp` - date and time comment created • `cmi.objectives.n.id` - 4000 char string - the objective identifier where "n" is the objective number • `cmi.objectives.n.score.scaled` - The objective score value scaled from -1.0 to 1.0

Call	Description
	• `cmi.objectives.n.score.raw` - The raw objective score value •cmi.objectives.n.score.max - The maximum possible objective score - if not set, assumed 100 • `cmi.objectives.n.score.min` - The minimum possible objective score - if not set, assumed 0 • `cmi.objectives.n.progress_measure` - Progress indicator - accepts a value from 0.0 to 1.0 • `cmi.objectives.n.success_status` - Indicates the learner has mastered the objective - accepts "passed", "failed", or "unknown" • `cmi.objectives.n.completion_status` - Indicates if the learner has completed the objective - accepts "completed", "incomplete", "not attempted", or "unknown" • `cmi.objectives.n.description` - 250 char string objective description • `cmi.learner_preference.audio_level` - Identifies learner preference for audio volume - accepts number > or = 0 • `cmi.learner_preference.language` - 250 char string identifying the learner language preference • `cmi.learner_preference.audio_captioning` - Enables captions - accepts "off", "no_change", or "on" • `cmi.learner_preference.deliverry_speed` - Identifies learner preference for presentation speed - accepts number > or = 0 • `cmi.interactions.n.id` - 4000 char string - accepts the interaction identifier where "n" is the interaction number • `cmi.interactions.n.type` - Accepts "true-false", "choice", "fill-in", "long-fill-in", "likert", "matching", "performance", "sequencing", "numeric", or "other" • `cmi.interactions.n.weighting` - Weight of the interaction for use in grading by the SCO • `cmi.interactions.n.learner_response` - Learner response - vocabulary depends on interaction type • `cmi.interactions.n.result` - Accepts "correct", "incorrect", "unanticipated", "neutral", or a numeric estimate of correctness • `cmi.interactions.n.latency` - Time (seconds) spent answering the interaction

Call	Description
	• `cmi.interactions.n.description` - 250 char string description of the interaction • `cmi.interfactions.n.timestamp` - Time the interaction was presented to the learner • `cmi.interactions.n.objectives.n.id` - 4000 char string - returns the objective identifier where "n" is the objective number • `cmi.interactions.n.correct_response.n.pattern` - Accepts a correct response pattern (answer) for the interaction - a single interaction may have multiple correct response patterns • `adl.nav.request` - Indicates the navigation request the LMS should call when the SCO is terminated - accepts "continue", "previous", "choice", "exit", "exitAll", "abandon", "abandonAll", or "_none_" The following are legitimate element name parameters, but are read-only. These would therefore return "false" and set an error: 404 - element is read-only. • `cmi._version` • `cmi.learner_id` • `cmi.learner_name` • `cmi.credit` • `cmi.completion_status` • `cmi.entry` • `cmi.completion_threshold` • `cmi.score._children` • `cmi.scaled_passing_score` • `cmi.total_time` • `cmi.mode` • `cmi.max_time_allowed` • `cmi.time_limit_action` • `cmi.launch_data` • `cmi.comments_from_learner._children` • `cmi.comments_from_learner._count` • `cmi.comments_from_lms.n.comment` • `cmi.comments_from_lms._children` • `cmi.comments_from_lms._count` • `cmi.comments_from_lms.n.location` • `cmi.comments_from_lms.n.timestamp`

Call	Description
	• `cmi.objectives._children` • `cmi.objectives._count` • `cmi.objectives.n.score_children` • `cmi.learner_preference._children` • `cmi.interactions._children` • `cmi.interactions._count` • `cmi.interactions.n.objectives._count` • `cmi.interactions.n.correct_response._count` • `adl.nav.request_valid.continue` • `adl.nav.request_valid.previous` • `adl.nav.request_valid.choice.{target=STRING}`
`Commit( )`	Any values set using the `SetValue` command are stored for delivery. This function submits any values not yet sent to the LMS.
`Terminate( )`	The content must call this function before it closes, if it successfully called Initialize at any point. It signals to the LMS that the content has finished communicating. The content may not call any API function except `GetLastError` after it calls `Terminate`
`GetLastError( )`	This function provides content with a way of assessing whether or not any given API call was successful, and if it was not successful, what went wrong. This routine returns an error code from the previous API call. Each time an API function is called (with the exception of this one), the error code is reset in the API. The content may call this any number of times to retrieve the error code, and the code will not change until the next API call.
`GetErrorString( )`	This function returns a textual description of the error represented by an error code number.
`GetDiagnostic( )`	This function would return an LMS-specific error description based on an error code number.

3.4 SCORM 2004 Sequencing

SCORM 2004 included a new sequencing specification that allows content authors to control the order in which SCOs, or content objects, are presented to a learner.

Sequencing allows learners to navigate through a series of content objects without having to return to the LMS to launch the objects. For complete documentation, please refer to SCORM Sequencing and Navigation with the Advanced Distributed Learning (ADL) initiative documentation.

The course structure and the sequencing definition are both defined in the `imsmanifest.xml` file. The course structure is defined by a grouping of `item` tags within an `organization` tag. An `item` tag which contains other `item` tags would be a folder in the course structure. Item tags which do not contain other items would be either a SCO or an asset in the structure. In sequencing terms, the organization tag would be the activity tree and the various item tags would be learning activities. The sequencing definition for a course structure sits within an organization tag and applies to various levels of the course structure.

3.4.1 Common SCORM Sequencing Terms

The SCORM 2004 sequencing standard introduced terminology to describe how sequencing rules apply.

In sequencing terms, a course structure is made up of *learning activities*, which are SCOs (content which uses SCORM communication), *assets* (content which does not use SCORM communication), or collections of content objects. These activities are structured into an *activity Tree*, which is effectively the same thing as the course structure. Activities which contain one or more sub-activities are called *clusters*. Specifically, a cluster consists of the parent activity and its immediate children. An activity which does not contain any sub-activities or children is a *leaf Activity*, and is either an SCO or asset.

Learners make *attempts* at an activity. An attempt is one effort to complete a given activity. An attempt is active as long as a learner is active in any of the activities sub-activities. For example, a learner could have an active attempts for the Course Root, Module 2, Lesson 05, and Chapter 1. An attempt at the root of the activity tree is a *sequencing session* and it is active as long as a learner is in the course.

Term	Definition
SCO	A content object that uses SCORM communication.
Asset	A content object that does not use SCORM communication.
Learning Activity / Activity	A generic name for a SCO, an Asset, or a group of SCOs, Assets, or other groups.
Cluster	A collection of learning activities made up of the parent activity and its immediate children.
Leaf Activity	A learning activity that does not contain sub-activities. This would be a SCO or an Asset.

Term	Definition
Activity Tree	A structure of learning activities. This is the same thing as the course structure and maps to an `organization` tag in the `imsmanifest.xml` file.
Attempt	An effort to complete an activity. An attempt is active as long a the learner is active in a sub-object of the activity.
Sequencing Session	An attempt at the root of the Activity Tree.

3.4.2 Additional Resources for SCORM Sequencing

SCORM sequencing is complex and requires further reading. We suggest these resources to understand SCORM sequencing in detail.

- The SCORM Sequencing and Navigation document provides the best source of information on the standard. The sequencing tags are also covered in the SCORM Content Aggregation Model document. These documents are available for download from the Advanced Distributed Learning (ADL) web site.
- Carnegie Mellon publishes a guide titled *SCORM Best Practices Guide for Content Developers* which includes a section on the sequencing standard including sequencing templates.
- ADL provides some sequencing samples on their web site that may be loaded into an LMS to demonstrate some common sequencing behaviors. Comparing the course behavior with sequencing information in the `imsmanifest.xml` files is a good way to dissect the sequencing standard.
- Reload Editor is a popular tool for generating manifest files with sequencing information.
- The forums on the ADL web site are a good location to find additional information on all aspects on the SCORM standards including information on third-party tools.

3.4.3 SCORM 2004 Sequencing Root Elements

There are two root elements for sequencing information: `sequencing` and `presentation`.

Root Element	Description
`sequencing`	Defined for `item` or `organization` elements. The sequencing element details all of the sequencing information for a given activity.
`presentation`	Defined for `item` elements that reference a SCO. The presentation element allows a course designer to remove navigation controls for a particular SCO.

3.4.4 `sequencing`

The `sequencing` element is a root level SCORM sequencing element.

- Prefix: `imsss:`
- Applies to Clusters and Leaf Activities
- Parent: None
- Children
  - `controlMode`
  - `sequencingRules`
  - `limitConditions`
  - `rollupRules`
  - `objectives`
  - `randomizationControls`
  - `constrainedChoiceConsiderations`
  - `rollupConsiderations`

Sample Code

```
<imsss:sequencing>
          <imsss:limitConditions attemptLimit="1"/>
          <imsss:rollupRules rollupObjectiveSatisfied="false"/>
</imsss:sequencing>
```

3.4.5 `controlMode`

The `controlMode` element applies rules to user navigation within a clusters for SCORM sequencing.

Attributes of `controlMode`

- `choice` (`true` or `false`): if true, children of the cluster are valid selections for a "Choice" navigation request allowing a learner to navigate to one of the cluster's children from another location in the course structure.
  - Applies to Clusters
  - Default: `true`
- `choiceExit` (`true` or `false`): indicates whether a "Choice" navigation request can target another activity that is not a descendent of this activity thereby causing the activity to terminate.
  - Applies to Clusters and leaves.
  - Default: `true`
- `flow` (`true` or `false`): indicates whether flow sequencing requests (previous or continue) are permitted to the children of this activity.
  - Applies to Clusters

    - Default: `false`
- `forwardOnly` (`true` or `false`): disables Previous requests, only allowing a learner to move forward through the children of the activity.
    - Applies to Clusters
    - Default: `false`
- `useCurrentAttemptObjInfo` (`true` or `false`): indicates that the objective progress information for the children of the activity will only be used in rule evaluations and rollup if that information was recorded during the current attempt on the activity.
    - Applies to Clusters
    - Default: `true`
- `useCurrentAttemptProgressInfo` (`true` or `false`): indicates that the attempt progress information for the children of the activity will only be used in rule evaluations and rollup if that information was recorded during the current attempt on the activity.
    - Applies to Clusters
    - Default: `true`

Details

- Prefix: `imsss:`
- Typically applies to Clusters, but some settings may apply to Leaf Activities
- Parent: `sequencing`
- Children: None

Sample Code

```
<imsss:sequencing>
     <imsss:controlMode choice="false" choiceExit="false" flow="true"
forwardOnly = "true"/>
 </imsss:sequencing>
```

3.4.6 `constrainedChoiceConsiderations`

Typically, a learner can navigate to any activity whose parent has a `controlMode` `choice` setting of `true`. The `constrainedChoiceConsiderations` element applies additional limits on `choice` navigation requests.

Attributes of `constrainedChoiceConsiderations`

- `constrainChoice`: this attribute indicates that only activities which are logically “next” from the constrained activity can be targets of a Choice navigation request. When true, it limits the Choice navigation request to activities that are immediately next or previous from the current activity.

- Applies to Clusters
- Default: `true`

- `preventActivation`: this attribute indicates that attempts on children activities should not begin unless the current activity is the parent. Prevents jumping to the contents of a cluster until the parent activity is active.
  - Applies to Clusters
  - Default: `true`

Details

Sample Code

```
<imsss:sequencing>
     <adlseq:constrainedChoiceConsiderations constrainChoice = "true" />
 </imsss:sequencing>
```

3.4.7 `sequencingRules`

The `sequencingRules` element is a container for SCORM sequencing rules.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencing`
- Children:
  - `preConditionRule`
  - `postConditionRule`
  - `exitConditionRule`

Sample Code

```
<imsss:sequencing>
           <imsss:sequencingRules>
             <imsss:preConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "disabled"/>
             </imsss:preConditionRule>
            </imsss:sequencingRules>
</imsss:sequencing>
```

3.4.8 `preConditionRule`

The `preConditionRule` element is SCORM a sequencing rule container that determines if an activity is presented to the learner.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencingRules`
- Children:
  - `ruleConditions`
  - `ruleAction`

Sample Code

```
<imsss:sequencing>
          <imsss:sequencingRules>
             <imsss:preConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "disabled"/>
             </imsss:preConditionRule>
          </imsss:sequencingRules>
</imsss:sequencing>
```

3.4.9 `postConditionRule`

The `postConditionRule` element is a SCORM sequencing rule container for rules to be processed when an activity terminates..

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencingRules`
- Children:
  - `ruleConditions`
  - `ruleAction`

Sample Code

```
<imsss:sequencing>
          <imsss:sequencingRules>
             <imsss:postConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "exitParent"/>
             </imsss:postConditionRule>
          </imsss:sequencingRules>
</imsss:sequencing>
```

3.4.10 `exitConditionRule`

The `postConditionRule` element is a SCORM sequencing rule container for rules to be processed when a descendent activity terminates

When applied to an activity, the rules are processed when any descendent activity terminates.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencingRules`
- Children:
  - `ruleConditions`
  - `ruleAction`

Sample Code

```
<imsss:sequencing>
      <imsss:sequencingRules>
           <imsss:exitConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "exit"/>
           </imsss:exitConditionRule>
      </imsss:sequencingRules>
 </imsss:sequencing>
```

3.4.11 `ruleConditions`

The `ruleConditions` element is a container for conditions which must be true for the rule action to be applied.

This is a sequencing rule container for the conditions which must be true for the rule action to be applied.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parents
  - `preConditionRule`
  - `postConditionRule`
  - `exitConditionRule`
- Children: `ruleCondition`

Sample Code

```
<imsss:sequencing>
      <imsss:sequencingRules>
           <imsss:exitConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "exit"/>
           </imsss:exitConditionRule>
      </imsss:sequencingRules>
```

```
</imsss:sequencing>
```

3.4.12 `ruleCondition`

The `ruleCondition` element is part of SCORM sequencing and describes a condition to be evaluated for a rule.

Attributes of `ruleCondition`

- `referencedObjective`: identifier of the Objective related to the activity and used in the evaluation of the rule condition. Type: string.
- `measureThreshold`: the value used as a threshold during measure-based condition evaluations. Number between -1.0000 to 1.0000.
- `operator`: the logical operator applied to the rule condition. Values can be:
  - `not` - The rule condition is negated during evaluation.
  - `noOp` (default) - The rule condition is processed as written during evaluation.
- `Condition` (required): the condition for the rule:
  - `satisfied` - True if Objective Progress Status and the Objective Satisfied Status are both true.
  - `objectiveStatusKnown` - True if Objective Progress Status is true.
  - `objectiveMeasureKnown` - True if Objective Progress Status and the Objective Measure Status are both true.
  - `objectiveMeasureGreaterThan` - True if Objective Progress Status is true and the Objective Normalized Measure is greater than the Rule Condition MeasureThreshold.
  - `completed` - True if Attempt Progress Status and the Attempt Completion Status for the activity are both true.
  - `activityProgressKnown` - True if Activity Progress Status and the Attempt Progress Status for the activity are both true.
  - `attempted` - True if Activity Progress Status is true and the Activity Attempt Count for the activity is positive.
  - `attemptLimitExceeded` - True if Activity Progress Status is true and the Limit Condition Attempt Limit Control attribute is true and the Activity Attempt Count for the activity is equal to or greater than the Limit Condition Attempt Limit.
  - `timeLimitExceeded`
  - `outsideAvailableTimeRange`
  - `always` (default) - Always True

Details

- Prefix: `imsss:`

- Applies to Clusters or Leaf Activities
- Parent
  - `ruleConditions`
- Children: None

Sample Code

```
<imsss:sequencing>
      <imsss:sequencingRules>
           <imsss:exitConditionRule>
                <imsss:ruleConditions>
                    <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "exit"/>
           </imsss:exitConditionRule>
      </imsss:sequencingRules>
 </imsss:sequencing>
```

3.4.13 `ruleAction`

The `ruleAction` element is the SCORM sequencing behavior if the parent rule evaluates true.

Note

The set of rule actions vary depending on the type of condition: `preConditionRule`, `postConditionRule`, or `exitConditionRule`).

Attributes of `ruleAction`

The `action` attribute is required. It identifies the sequencing behavior if rule evaluates to true. The potential values of `action` depend on the type of condition rules.

- Values for PreConditionRules:
  - `skip` - The activity is not considered a candidate for delivery during a "Flow" sequencing request.
  - `disabled` - The activity cannot be a target of any sequencing or delivery request.
  - `hiddenfromChoice` - The activity cannot be a target of a "Choice" sequencing request.
  - `stopForwardTraversal` - Any activities that follow the specified activity will not be considered candidates for delivery.
- Values for PostConditionRules:
  - `exitParent` - Process an "Exit Parent" termination request.
  - `ExitAll` - Process an "Exit All" termination request and return a "Exit" sequencing request.
  - `Retry` - Return a "Retry" sequencing request.
  - `retryAll` - Process an "Exit All" termination request and return a "Start" sequencing request.
  - `continue` - Return a "Continue" sequencing request
  - `previous` - Return a "Previous" sequencing request

- Values for ExitConditionsRules:
  - `exit` - Unconditionally Terminate the activity.

Details

Sample Code

```
<imsss:sequencing>
      <imsss:sequencingRules>
           <imsss:exitConditionRule>
                <imsss:ruleConditions>
                     <imsss:ruleCondition condition = "satisfied"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action = "exit"/>
           </imsss:exitConditionRule>
      </imsss:sequencingRules>
 </imsss:sequencing>
```

3.4.14 `limitConditions`

The `limitConditions` element defines conditions under which an activity is cannot be delivered. When a limit condition is met or exceeded, the activity becomes unavailable for delivery.

Attributes of `limitConditions`

- `attemptLimit` (integer) - Identifies the maximum number of attempts allowed for the activity.
- `attemptAbsoluteDurationLimit` (#.# seconds) - Identifies the maximum time duration that a learner is permitted to spend on a single attempt of the activity. This element is used to initialize the `cmi.max_time_allowed` in the SCORM RTE.

Details

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencing`
- Children: None

Sample Code

```
<imsss:sequencing>
      <imsss:limitConditions attemptLimit="1"/>
```

```
</imsss:sequencing>
```

3.4.15 `rollupRules`

The `rollupRules` element is a container for a set of SCORM sequencing rules used to roll up learner progress to a cluster.

Attributes of `rollupRules`

- `rollupObjectiveSatisfied`: indicates that the objective's satisfied status associated with the activity is included in the rollup for its parent activity. Values can be `true` or `false`. Defaults to `true`.
- `rollupProgressCompletion`: indicates that the attempt's completion status associated with the activity is included in the rollup for its parent activity. Values can be `true` or `false`. Defaults to `true`.
- `objectiveMeasureWeight`: indicates the weighting factor applied to the objectives normalized measure is used during rollup for the parent activity. Can be a number 0.0000 to 1.0000.

Details

→ Tip

You can think of `rollupRules` as a logical statement: for a given cluster, if some conditions are true for some subset of child activities then some action is applied to the parent activity.

- Prefix: `imsss:`
- Applies to Clusters
- Parent: `sequencing`
- Children: `rollupRule`

{·} Sample Code

```
<imsss:sequencing>
      <imsss:rollupRules >
            <imsss:rollupRule childActivitySet = "all">
                  <imsss:rollupConditions>
                        <imsss:rollupCondition condition = "attempted"/>
                  </imsss:rollupConditions>
                  <imsss:rollupAction action = "completed"/>
            </imsss:rollupRule>
      </imsss:rollupRules>
 </imsss:sequencing>
```

3.4.16 `rollupRule`

The `rollupRule` element is part of SCORM sequencing and a container for an individual rule used to roll up learner progress to a cluster.

Attributes of `rollupRule`

- `childActivitySet`: indicates whose data values are used to evaluate the rollup condition. Possible values:
  - `all`: apply the rollup action if the conditions are true for all the children of the cluster. This is the default value.
  - `any`: apply the rollup action if the conditions are true for any of the children of the cluster.
  - `none`: apply the rollup action if the conditions are false for all the children of the cluster.
  - `atLeastCount`: apply the rollup action if the conditions are true for at least the number of children specified in the `minimumCount` attribute.
  - `atLeastPercent` - Apply the rollup action if the conditions are true for at least the percentage of children specified in the `minimumPercentage` attribute.
- `minimumCount`: the `minimumCount` attribute shall be used when the `childActivitySet` attribute is set to `atLeastCount`. The rollup rule condition evaluates to true if at least the number of children specified by this attribute have a rollup condition of true. This is an integer.
- `minimumPercent`: the `minimumPercent` attribute shall be used when the `childActivitySet` attribute is set to `atLeastPercent`. The rollup rule condition evaluates to true if at least the percentage of children specified by this attribute have a rollup condition value of true. Can be a number 0.0000 to 1.0000.

Details

> → Tip
>
> You can think of `rollupRules` as a logical statement: for a given cluster, if some conditions are true for some subset of child activities then some action is applied to the parent activity.

- Prefix: `imsss:`
- Applies to Clusters
- Parent: `rollupRules`
- Children:
  - `rollupConditions`
  - `rollupAction`

> {·} Sample Code

```
<imsss:sequencing>
      <imsss:rollupRules >
           <imsss:rollupRule childActivitySet = "all">
                <imsss:rollupConditions>
                     <imsss:rollupCondition condition = "attempted"/>
```

```
                </imsss:rollupConditions>
                <imsss:rollupAction action = "completed"/>
            </imsss:rollupRule>
        </imsss:rollupRules>
    </imsss:sequencing>
```

3.4.17 `rollupConditions`

The `rollupConditions` element is part of SCORM sequencing and a container for a set of conditions used to evaluate a rollup rule.

Attributes of `rollupConditions`

- `ConditionCombination`: indicates how the rollup conditions are combined.
  - `all`: the condition set is true if **all** of its conditions are true.
  - `any`: the condition set is true if **any** of its conditions are true. This is the default value.

Details

- Prefix: `imsss:`
- Applies to Clusters
- Parent = `rollupRule`
- Children = `rollupCondition`

Sample Code

```
<imsss:sequencing>
      <imsss:rollupRules >
            <imsss:rollupRule childActivitySet = "all">
                  <imsss:rollupConditions>
                        <imsss:rollupCondition condition = "attempted"/>
                  </imsss:rollupConditions>
                  <imsss:rollupAction action = "completed"/>
            </imsss:rollupRule>
      </imsss:rollupRules>
 </imsss:sequencing>
```

3.4.18 `rollupCondition`

The `rollupCondition` element is part of SCORM sequencing and a container for an individual condition used in evaluating a rollup rule.

Attributes of `rollupCondition`

- `operator`: the logical operator applied to the rule condition. Values:
  - `not`: the rule condition is negated during evaluation.
  - `noOp`: the rule condition is processed as written during evaluation. This is the default value.
- `Condition`: the condition evaluated. Required. Values:
  - `satisfied`: true if Objective Progress Status and the Objective Satisfied Status are both true for the child activity.
  - `objectiveStatusKnown`: true if Objective Progress Status is true for the child activity.
  - `objectiveMeasureKnown`: true if Objective Progress Status and the Objective Measure Status are both true for the child activity.
  - `completed`: true if Attempt Progress Status and the Attempt Completion Status for the child activity are both true.
  - `activityProgressKnown`: true if Activity Progress Status and the Attempt Progress Status for the child activity are both true.
  - `attempted`: true if Activity Progress Status is true and the Activity Attempt Count for the child activity is positive.
  - `attemptLimitExceeded`: true if Activity Progress Status is true and the Limit Condition Attempt Limit Control attribute is true and the Activity Attempt Count for the child activity is equal to or greater than the Limit Condition Attempt Limit.
  - `timeLimitExceeded`
  - `outsideAvailableTimeRange`
  - `Never`: always false

Details

- Prefix: `imsss:`
- Applies to Clusters
- Parent: `rollupConditions`
- Children: none

Sample Code

```
<imsss:sequencing>
      <imsss:rollupRules >
           <imsss:rollupRule childActivitySet = "all">
                <imsss:rollupConditions>
                     <imsss:rollupCondition condition = "attempted"/>
```

```
                </imsss:rollupConditions>
                <imsss:rollupAction action = "completed"/>
            </imsss:rollupRule>
        </imsss:rollupRules>
 </imsss:sequencing>
```

3.4.19 `rollupAction`

The `rollupAction` element is part of SCORM sequencing and identifies the condition to be applied when the rollup rule evaluates true.

Attributes of `rollupAction`

- `Action`: the action to be taken if the rollup rule evaluates to true. Required. Values:
  - `satisfied`: sets the Objective Progress Status and the Objective Satisfied Status for the rolled up objective to true.
  - `notSatisfied`: sets the Objective Progress Status for the rolled up objective to true and the Objective Satisfied Status for the rolled up objective to false.
  - `Completd`: sets the Attempt Progress Status and the Attempt Completion Status for the rolled up objective to true.
  - `incomplete`: sets the Attempt Progress Status for the rolled up objective to true and the Attempt Completion Status for the rolled up objective to false.

Details

- Prefix: `imsss:`
- Applies to Clusters
- Parent: `rollupRule`
- Children: None

Sample Code

```
<imsss:sequencing>
      <imsss:rollupRules >
            <imsss:rollupRule childActivitySet = "all">
                <imsss:rollupConditions>
                     <imsss:rollupCondition condition = "attempted"/>
                </imsss:rollupConditions>
                <imsss:rollupAction action = "completed"/>
            </imsss:rollupRule>
      </imsss:rollupRules>
 </imsss:sequencing>
```

3.4.20 `rollupConsiderations`

The `rollupConsiderations` element is part of SCORM sequencing and used to further refine the conditions under which an activity contributes to the rollup of its parent.

Attributes of `rollupConsiderations`

- `RequiredForSatisfied`: indicates the condition under which the activity is included in its parent's evaluation of a satisfied rollup rule
- `RequiredForNotSatisfied`: indicates the condition under which the activity is included in its parent's evaluation of a not satisfied rollup rule.
- `RequiredForCompleted`: indicates the condition under which the activity is included in its parent's evaluation of a completed rollup rule.
- `RequiredForIncomplete`: indicates the condition under which the activity is included in its parent's evaluation of a incomplete rollup rule.
- `measureSatisfactionIfActive`: indicates if the measure should be used to determine satisfaction during rollup when the activity is active. Can be `true` or `false`. Default is `false`.

Values of `rollupConsiderations` Attributes

The following values apply to all of the above attributes except `measureSatisfactionIfActive`:

- `always`: the activity is always included in the rollup rule processing. This is the default value.
- `ifAttempted`: the activity is included in the rollup processes if the activity was attempted.
- `ifNotSkipped`: the activity is included in the rollup processes if the activity was not skipped
- `ifNotSuspended`: the activity is included in the rollup processes if the activity was not suspended.

Details

Rollup rules are applied at the cluster level. The rollup considerations allow for refinement at the child activity level.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencing`
- Children: None

Sample Code

```
<imsss:sequencing>
      <adlseq:rollupConsiderations measureSatisfactionIfActive = "false"
requiredForCompleted = "ifNotSkipped" />
 </imsss:sequencing>
```

3.4.21 `objectives`

The `objectives` element is part of SCORM sequencing and is a container for the set of objectives associated with an activity.

Attributes of `objectives`

None

Details

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `sequencing`
- Children:
  - `primaryObjective`
  - `objective`

Sample Code

```
<imsss:sequencing>
      <imsss:objectives>
            <imsss:primaryObjective objectiveID = "PRIMARYOBJ" satisfiedByMeasure
= "true">
                  <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
                  <imsss:mapInfo targetObjectiveID = "obj_module_1"
readNormalizedMeasure = "false" writeSatisfiedStatus = "true" />
            </imsss:primaryObjective>
     </imsss:objectives>
</imsss:sequencing>
```

3.4.22 `primaryObjective`

The `primaryObjective` element is part of SCORM sequencing and identifies the objective that contributes to the rollup associated with the activity.

Attributes of `primaryObjective`

- `satisfiedByMeasure`: if true, the `minNormalizedMeasure` value is used to determine if the objective is satisfied. Values can be `true` or `false`. Default is `false`.
- `objectiveID`: the unique identifier for the objective. This attribute is optional unless a `mapInfo` element is present for the primary objective.

Details

If the `objectives` element is defined, then the `primaryObjective` is mandatory even if it is empty. The `objectives` element may only contain **one** primary objective element.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `objectives`
- Children:
  - `minNormalizedMeasure`
  - `mapInfo`

Sample Code

```
<imsss:sequencing>
      <imsss:objectives>
           <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
satisfiedByMeasure = "true">
                <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
                <imsss:mapInfo targetObjectiveID = "obj_module_1"
readNormalizedMeasure = "false" writeSatisfiedStatus = "true" />
           </imsss:primaryObjective>
     </imsss:objectives>
</imsss:sequencing>
```

3.4.23 `objective`

The `objective` element is part of SCORM sequencing and identifies an objective that does not contribute to the rollup associated with the activity.

Attributes of `objective`

- `satisfiedByMeasure`: if true, the `minNormalizedMeasure` value is used to determine if the objective is satisfied. Values can be `true` or `false`. Default is `false`.
- `objectiveID`: the unique identifier for the objective. Required.

Details

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent: `objectives`
- Children:
  - `minNormalizedMeasure`
  - `mapInfo`

Sample Code

```
<imsss:sequencing>
      <imsss:objectives>
           <imsss:primaryObjective objectiveID = "PRIMARYOBJ" />
           <imsss:objective objectiveID="obj_module_1">
                <imsss:mapInfo targetObjectiveID="obj_module_1"
readSatisfiedStatus = "false" readNormalizedMeasure = "false"
writeSatisfiedStatus = "true" />
           </imsss:objective>
     </imsss:objectives>
</imsss:sequencing>
```

3.4.24 `minNormalizedMeasure`

The `minNormalizedMeasure` element is part of SCORM sequencing and identifies the minimum satisfaction measure for the objective.

Attributes of `minNormalizedMeasure`

None

Details

The value of `minNormalizedMeasure` is normalized between –1.0 and 1.0. The value is inclusive. The default is 1.0.

If `minNormalizedMeasure` defines a minimum satisfaction measure for the **primary** objective (the `primaryObjective` element), then the Learning Management System (LMS) shall use this value to initialize the `cmi.scaled_passing_score` in the SCORM RTE.

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent:
  - `objective`
  - `primaryObjective`
- Children: None

Sample Code

```
<imsss:sequencing>
      <imsss:objectives>
           <imsss:primaryObjective objectiveID = "PRIMARYOBJ" satisfiedByMeasure
= "true">
                <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
                <imsss:mapInfo targetObjectiveID = "obj_module_1"
readNormalizedMeasure = "false" writeSatisfiedStatus = "true" />
           </imsss:primaryObjective>
     </imsss:objectives>
</imsss:sequencing>
```

3.4.25 `mapInfo`

The `mapInfo` element is part of SCORM sequencing and maps an objective that is local to the activity to a shared global objective.

Attributes of `mapInfo`

- `targetObjectiveID`: the unique identifier for the global shared objective. Required.
- `readSatisfiedStatus`: indicates that when the satisfaction status for the local objective is undefined, that status should be read from the global shared objective. Values can be `true` or `false`. Default is `true`.
- `readNormalizedMeasure`: indicates that when the normalized measure for the local objective is undefined, that measure should be read from the global shared objective. Values can be `true` or `false`. Default is `true`.
- `writeSatisfiedStatus`: indicates that the satisfaction status for the local objective should be written to the shared global objective upon termination of the activity. Values can be `true` or `false`. Default is `false`.
- `writeNormalizedMeasure`: indicates that the normalized measure for the local objective should be written to the shared global objective upon termination of the activity. Values can be `true` or `false`. Default is `false`.

Details

- Prefix: `imsss:`
- Applies to Clusters or Leaf Activities
- Parent:
  - `objective`
  - `primaryObjective`
- Children: none

Sample Code

```
<imsss:sequencing>
      <imsss:objectives>
            <imsss:primaryObjective objectiveID = "PRIMARYOBJ" satisfiedByMeasure
= "true">
                  <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
                  <imsss:mapInfo targetObjectiveID = "obj_module_1"
readNormalizedMeasure = "false" writeSatisfiedStatus = "true" />
            </imsss:primaryObjective>
     </imsss:objectives>
</imsss:sequencing>
```

3.4.26 `randomizationControls`

The `randomizationControls` element is part of SCORM sequencing and describes how children of an activity should be ordered during the sequence process.

Attributes of `randomizationControls`

- `reorderChildren`: indicates the order of child activities is randomized. Values can be `true` or `false`. Default is `true`.
- `randomizationTiming`: indicates when the `reorderChildren` attribute should occur. Values:
  - `never`: never applied. Default.
  - `once`: applied before the first attempt on the activity.
  - `onEachAttempt`: applied before each new attempt on the activity.
- `selectCount`: the number of child activities that must be selected from the set of child activities associated with the activity. An integer.
- `selectionTiming`: indicates when the `selectCount` attribute should occur.
  - `never`: never applied. Default.
  - `once`: applied before the first attempt on the activity.
  - `onEachAttempt`: applied before each new attempt on the activity.

Details

> → Tip
>
> You can use `randomizationControls` to write a rule for sequencing behavior. For example: select 4 of the 6 activities on the first attempt of an activity.

- Prefix: `imsss:`
- Applies to Clusters
- Parent: `sequencing`
- Children: none

> {·} Sample Code

```
<imsss:sequencing>
      <imsss:randomizationControls selectCount="2"
selectionTiming="onEachNewAttempt" />
 </imsss:sequencing>
```

3.4.27 `deliveryControls`

The `deliveryControls` element is part of SCORM sequencing and describes actions that the Learning Management System (LMS) should take before and after an attempt on the activity.

Attributes of `deliveryControls`

- `Tracked`: indicates that the objective progress information and activity/attempt progress information for the attempt should be recorded and the data will contribute to the rollup for its parent activity. Values can be `true` or `false`. Default is `true`.
- `completionSetByContent`: indicates that the attempt completion status for the activity will be set by the SCO. Values can be `true` or `false`. Default is `false`.
- `objectiveSetByContent`: indicates the objective satisfied status for the activity's associated objective that contributes to rollup is set by the SCO. Values can be `true` or `false`. Default is `false`.

> ⓘ Note
>
> When `completionSetByContent` and `objectiveSetByContent` are `false`, SCORM 2004 4th edition content can sometimes be unexpectedly marked as complete when learners navigate away from or exit their learning content. To bypass the `false` values of the settings in the course manifest so content completion is recorded only after the content communicates completion to the Learning system, you can set `contentCommunicatesCompletion` to `true` in *Learning Administration* > *System Administration* > *Configuration* > *System Configuration* > *LMS_ADMIN*.
>
> Please note that when `contentCommunicatesCompletion=true`, content within the course that doesn't communicate with the LMS for completion (such as images or documents) won't be tracked by Learning as

before, which can prevent learner progress on these courses. To avoid this issue, we recommend ensuring that non-communicating assets are not marked as required for completion of the affected course.

Details

- Prefix: `imsss:`
- Applies to Leaf Activities
- Parent: `sequencing`
- Children: none

Sample Code

```
<imsss:sequencing>
      <imsss:deliveryControls tracked = "false"/>
 </imsss:sequencing>
```

3.4.28 `presentation`

The `presentation` element part of SCORM sequencing and is a container element that encapsulates presentation information for a given learning activity.

Attributes of `presentation`

None

Details

- Prefix: `adlnav:`
- Applies to Leaf Activities
- Parent: None
- Children: `navigationInterface`

Sample Code

```
<adlnav:presentation>
      <adlnav:navigationInterface>
           <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
           <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
</adlnav:presentation>
```

3.4.29 `navigationInterface`

The `navigationInterface` element is part of SCORM sequencing and a container element that encapsulates presentation information for a given learning activity.

Attributes of `navigationInterface`

None

Details

- Prefix: `adlnav:`
- Applies to Leaf Activities
- Parent: `presentation`
- Children: `hideLMSUI`

Sample Code

```
<adlnav:presentation>
      <adlnav:navigationInterface>
           <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
           <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
</adlnav:presentation>
```

3.4.30 `hideLMSUI`

The `hideLMSUI` element is part of SCORM sequencing and indicates that the Learning Management System (LMS) **shouldn't** provide user interface devices that enable the learner to trigger specific events. It allows an activity to hide LMS navigation buttons.

Attributes of `hideLMSUI`

None

Values of `hideLMSUI`

> **Note**
>
> Depending on how you or the LMS defines the buttons and depending on the language of the user, the exact names of the buttons can differ.

- `previous`: hides the *Previous* button.
- `continue`: hides the *Continue* button.
- `exit`: hides the *Exit* button.
- `exitAll`: hides the *Quit* button (SCORM 2004 4th edition only).
- `suspendAll`: hides the *Suspend* button (SCORM 2004 4th edition only).

> **Note**
>
> It is not recommended to hide all navigational buttons if your content isn't making its own navigation requests.

Details

- Prefix: `adlnav:`
- Applies to Leaf Activities
- Parent: *navigationInterface*
- Children: none

> **Sample Code**

```
<adlnav:presentation>
      <adlnav:navigationInterface>
           <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
           <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
 </adlnav:presentation>
```

Related Information

SAP SuccessFactors Learning Navigation Controls for SCORM 2004 Content [page 47]
SAP SuccessFactors Learning Navigation Controls for SCORM 2004 Content [page 47]
Structured Content Troubleshooting [page 73]

3.4.31 SAP SuccessFactors Learning Navigation Controls for SCORM 2004 Content

SAP SuccessFactors Learning navigation buttons for SCORM 2004 follow SCORM 2004 sequencing rules.

In SAP SuccessFactors Learning, SCORM content appears in a wrapper. When the learner views SCORM 2004 content in the wrapper page, the buttons on the wrapper page follow SCORM 2004 sequencing rules. The sequencing rules for the course dictate the navigation wrapper's appearance and behavior. The following buttons can appear in the wrapper navigation bar:

- *Suspend* allows learners to pause the course. The next launch resumes the same attempt.
- *Previous* allows learners to navigate to the previous content object in the course sequence.
- *Continue* allows learners to navigate to the next content object in the course sequence.
- *Exit* allows learners to end the current attempt.
  - When learners choose *Exit*, they exit completely from the content.
  - Unlike *Suspend*, the next time a learner starts the content, SAP SuccessFactors Learning recognizes it as a new attempt.
- *Quit* has the same functionality as the *Exit* button and appears only in SCORM 2004 4th edition.

In SCORM, the course structure is defined in the imsmanifest file. The manifest details how many content objects are in the course and how they're structured in relation to one another. The sequencing standard sits on top of this structure and defines how a learner may proceed through the course. For example, the sequencing rules may dictate that a learner must complete the content objects in a linear order.

> **Note**
>
> Terms like attempt have a specific meaning in SCORM sequencing.

Related Information

Structured Content Troubleshooting [page 73]

3.5 SCORM API

A SCORM compliant Learning Management System (LMS) like SAP SuccessFactors Learning provides an Application Programming Interface (API) that understands a set of JavaScript functions that are defined by the SCORM standard.

The API handles the communication with the LMS so that SCORM content can communicate with any SCORM-compliant LMS by using the predefined API functions. Using SCORM, therefore, requires understanding the API calls that the LMS supports and how to configure your LMS to call the API.

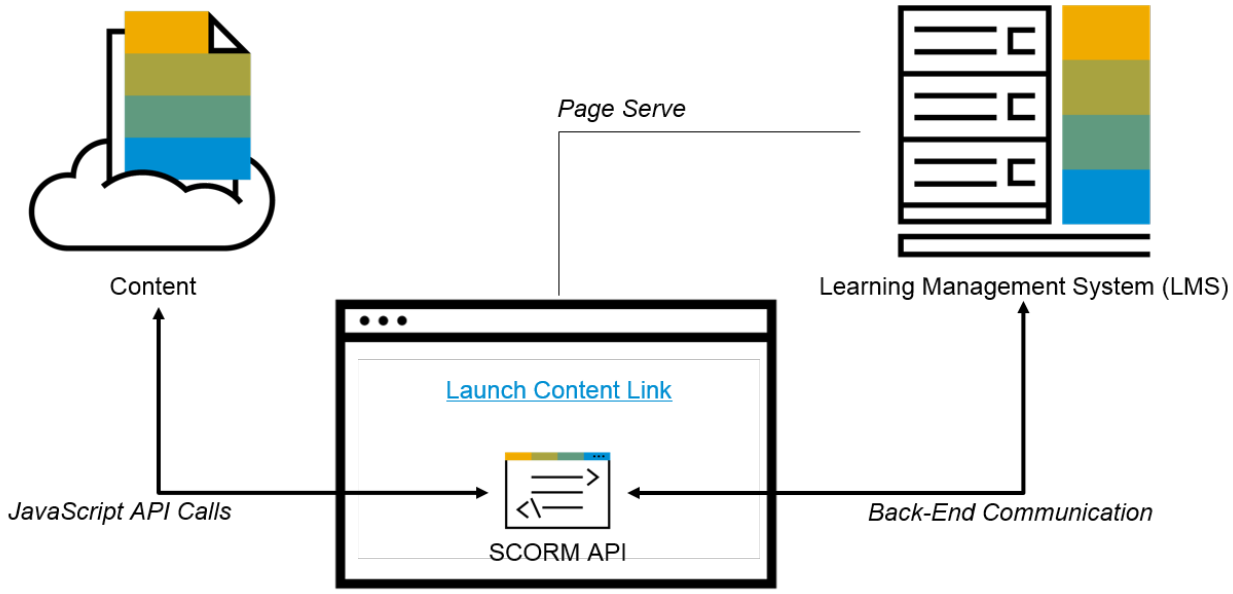
One major configuration roadblock is cross-domain hosting. If SCORM content isn't hosted under the same Domain Name Servers (DNS) (someserver.yourcompany.com) as the LMS, the browser prevents the SCORM

Content Object (SCO) from communicating with the API. SAP SuccessFactors Learning offers a cross-domain solution to enable a configuration under different DNS.

> **Note**
>
> Between SCORM 1.2 and SCORM 2004, the SCORM API was renamed from `API` to `API_1484_11`.

SCORM communication is processed by a client-side API provided by the LMS. The Sharable Content Object (SCO) communicates with the API through a series of SCORM-defined JavaScript function calls. The API processes the functions and handles the communication with the LMS.

1. The SCORM content searches itself for the API.
2. If it can't find the API in itself, it searches the opener window, which is where the API is for SAP SuccessFactors Learning exists: in the opener window served by the LMS.
3. When the API is found, the content initializes the API, and then gets and sets values to run the SCORM communication.
4. When finished, the content closes the communication with the API.



SCORM API Overview

Related Information

Proxlet Cross Domain Solution Overview [page 95]

3.6 SCORM 1.2

Sharable Content Object Reference Model (SCORM) 1.2 is an early release of the SCORM standard and is divided into three books.

Each of these books is available from the founders of the SCORM standard, Advanced Distributed Learning (ADL), which is an initiative for the Department of Defense.

- Overview: Covers areas of the CAM, RTE, and SN books at a high level.
- Content Aggregation Model (CAM): The SCORM Content Aggregation Model contains guidance for identifying and aggregating resources into structured learning content. This standard describes the content of the imsmanifest.xml file which details the course structure, supporting meta-data and sequencing information (SCORM 2004 only). The imsmanifest.xml is the file used by an LMS to import SCORM content.
- Run Time Environment (RTE): The SCORM Run Time Environment includes guidance for launching, communicating with and tracking content in a web-based environment. This book was originally derived from the run-time environment functionality defined in AICC's CMI001 Guidelines for Interoperability. It details the responsibilities of both the content and the LMS regarding the communication of tracking data.

3.6.1 SCORM 1.2 API Calls Supported by SAP SuccessFactors Learning

SAP SuccessFactors Learning supports a set of SCORM 1.2 API calls.

Call	Description
`LMSInitialize( )`	The content must call this function before calling any other API function. It indicates to the LMS system that the content is going to communicate. The LMS can take any initialization steps required in this function. For instance, this can contact the CMI via AICC messaging to pull down the `cmi.core` variables for quick access.

Call	Description
`LMSGetValue( )`	This function is used to pass data from the CMI to the content. Only one value is returned for each call. The category and/or element is named in the parameter. The following LMSGetValue parameters are supported: • `cmi.core._children` - returns "student_id, student_name, lesson_location, credit, lesson_status, entry, score, total_time, exit, session_time" • `cmi.core.student_id` • `cmi.core.student_name` • `cmi.core.lesson_location` - 255 char string typically used for bookmarking • `cmi.core.credit` - returns "no-credit" or "credit" • `cmi.core.lesson_status` - indicates SCO completion - returns one of the following: "passed", "completed", "failed", "incomplete", "browsed", or "not attempted" • `cmi.core.entry` - returns "ab-initio", "resume", or "" • `cmi.core.score._children` - returns "raw, min, max" • `cmi.core.score.raw` - the raw score value • `cmi.core.score.max` - the maximum possible score - if not set, assumed 100 • `cmi.core.score.min` - the minimum possible score - if not set, assumed 0 • `cmi.core.total_time` - returns the sum of all a learner's session times • `cmi.core.lesson_mode` - returns "browse", "normal", or "review" • `cmi.suspend_data` - 64000 char string usually used for progress data • `cmi.launch_data` - 4096 char string used to pass data to the SCO on launch from the LMS • `cmi.objectives._children` - returns "id, score, status" • `cmi.objectives._count` - returns the number of objectives currently stored for this SCO. • `cmi.objectives.n.id` - returns the objective identifier where "n" is the objective number • `cmi.objectives.n.score._children` - returns "raw, min, max " • `cmi.objectives.n.score.raw` - the raw objective score value

Call	Description
	• `cmi.objectives.n.score.min` - The minimum possible score - if not set, assumed 0 • `cmi.objectives.n.score.max` - the maximum possible objective score - if not set, assumed 100 • `cmi.objectives.n.status` - indicates if the learner has completed the objective - returns one of the following: "passed", "completed", "failed", "incomplete", "browsed", or "not attempted" The following are legitimate element names, but are write-only. These would therefore return "" and set an error: 404 - element is write-only. • `cmi.core.exit` • `cmi.core.session_time`

Call	Description
`LMSSetValue( )`	This function is used to pass data from the content to the LMS. The parameter indicates which category or element is being set. Only one value may be set with a single function call. The following `LMSSetValue` parameters are supported: • `cmi.core.lesson_location` - 255 char string usually used for bookmarking • `cmi.core.lesson_status` - indicates SCO completion - accepts "passed", "completed", "failed", "incomplete", "browsed", "not attempted" • `cmi.core.score.raw` - the raw score value • `cmi.core.score.max` - the maximum possible score - if not set, assumed 100 • `cmi.core.score.min` - the minimum possible score - if not set, assumed 0 • `cmi.core.exit` - indicates how the learner left the SCO - accepts "time-out", "suspend", or "logout" • `cmi.core_session_time` - Time spent in the SCO - HHHH:MM:SS.SS • `cmi.suspend_data` - 64000 char string usually used for progress data • `cmi.objectives.n.id` - the objective identifier where "n" is the objective number • `cmi.objectives.n.score.raw` - the raw objective score value • `cmi.objectives.n.score.min` - The minimum possible score - if not set, assumed 0 • `cmi.objectives.n.score.max` - the maximum possible objective score - if not set, assumed 100 • `cmi.objectives.n.status` - indicates if the learner has completed the objective - accepts "passed", "completed", "failed", "incomplete", "browsed", or "not attempted" The following are legitimate element names, but are read-only. These would therefore return "false" and set an error: 403 - element is read-only: • `cmi.core.student_id` • `cmi.core.student_name` • `cmi.core.credit` • `cmi.core.entry` • `cmi.core.total_time`

Call	Description
	• `cmi.core.lesson_mode` • `cmi.launch_data` The following elements are legitimate element names, but are keywords for multiple elements. These would therefore return "" and set the error code 402 - "Invalid set value, element is a keyword": • `cmi.core._children` • `cmi.core.score._children` • `cmi.objectives._children` • `cmi.objectives._count` • `cmi.objectives.n.score._children`
`LMSCommit( )`	Any values set using the `LMSSetValue` command are stored for delivery. This function submits any values not yet sent to the LMS.
`LMSFinish( )`	The content must call this function before it terminates, if it successfully called `LMSInitialize` at any point. It signals to the LMS that the content has finished communicating. The content may not call any API function except `LMSGetLastError` after it calls `LMSFinish`.
`LMSGetLastError( )`	This function provides content with a way of assessing whether or not any given API call was successful, and if it was not successful, what went wrong. This routine returns an error code from the previous API call. Each time an API function is called (with the exception of this one), the error code is reset in the API. The content may call this any number of times to retrieve the error code, and the code will not change until the next API call.
`LMSGetErrorString( )`	This function returns a textual description of the error represented by an error code number.
`LMSGetDiagnostic( )`	This function would return an LMS-specific error description based on an error code number.

4 AICC Overview

Although we can give you a brief overview of AICC, we strongly recommend that if you manage your own deployments of AICC, that you read the documentation to fully understand AICC.

The Aviation Industry CBT (Computer-Based Training) Committee (AICC) was an international association of technology-based training professionals. The AICC developed guidelines for the aviation industry in the development, delivery, and evaluation of CBT and related training technologies.

AICC provided a standard for communication between training content and Learning Management Systems (LMS).

> ⓘ Note
>
> In December 2014, the AICC formally announced that it had dissolved citing declining participation. The AICC transferred the CMI-5 effort and its document archive to the ADL.

The LMS is focused on two facets of the AICC guidelines.

Communication Between Content and an LMS

The Communication guidelines set well-defined rules for passing information from content to an LMS and vice versa. The guidelines specify the method by which information should be passed, and the format in which the information should be communicated (analogous to saying we'll communicate by telephone and in French). The AICC outlined two means of communicating with online content. These are defined in Appendix A and Appendix B of the AICC CMI Guidelines for Interoperability.

> ⓘ Note
>
> SAP SuccessFactors Learning fully supports the communication method outline in Appendix A (also known as HACP communication). SAP SuccessFactors also supports the Appendix B communication method as implemented under SCORM 1.2.

Content Setup Within an LMS

The AICC guidelines define a set of files used to provide an LMS with course information. These files can be used by an LMS to import the course. It's important to note that the actual content isn't imported or exported, just the definition or setup information for the content; the item (component) and content object definitions.

4.1 The Difference Between AICC Appendix A and Appendix B

Appendix A and B share a common data model, but differ in their communication process.

AICC outlines two communication methods in appendix A and appendix B of the *AICC CMI Guidelines for Interoperability* document. The two methods share a common data model but they use a different method to communicate information.

- Appendix A / HACP uses a direct HTTP post to pass information to, and retrieve information from, an LMS. In SAP SuccessFactors Learning, the HTTP requests are processed by a servlet running on the LMS server. Appendix A is also known as the HACP (HTTP AICC Communication Protocol) method.
- Appendix B / API uses a client-side API, provided by SAP SuccessFactors Learning, to process function calls from the content. The API then communicates to the LMS for the content. This method is far easier to implement for content providers since they do not need to construct HTTP posts, they are simply calling predefined functions understood by the LMS-provided API which serves as a go-between for the content and the LMS. Appendix B is sometimes referred to as the API method. SAP SuccessFactors Learning implements this communication method under SCORM 1.2.

4.2 SAP SuccessFactors Learning AICC Support

SAP SuccessFactors Learning supports a limited standard of AICC.

SAP SuccessFactors Learning versions 3.4 and above support AICC Appendix A (HACP) technology.

SAP SuccessFactors Learning versions 4.1 and above support AICC Appendix B technology as implemented under the SCORM 1.2 standard (see the SCORM section of this guide).

4.3 AICC Data Format

AICC communication is performed by HTTP posts comprised of a series of name/value pairs

When AICC content is launched from a Learning Management System (LMS), two parameters are appended to the launch url: `AICC_SID` and `AICC_URL`. The content retrieves these parameters from the launch path and uses them to communicate back to the LMS (SAP SuccessFactors Learning). The `AICC_URL` is the URL to which AICC data is posted. The `AICC_SID` is a unique identifier for the learner, course, and content object. In an AICC post, the `AICC_SID` is returned to the LMS under the `session_id` parameter.

In SAP SuccessFactors Learning, the AICC launch parameters are composed of the following data:

- `AICC_URL = <LMS protocol>://<LMS DNS>:<LMS port>/<LMS application context>/PwsAicc`: for example if a learner signed in to SAP SuccessFactors using `https://lms.company.com:7008/plateau/user/login.jsp` then the `AICC_URL` is `https://lms.company.com:7008/plateau/PwsAicc`.
- `AICC_SID` is an encrypted string of the user's LMS `session_id`.

AICC HTTP Post Parameters

AICC Parameter	Description
`command`	This can be any valid AICC HTTP command (for example, `PutParam` or `GetParam`).
`version`	AICC Spec Vision (SAP SuccessFactors Administration is certified under version 2.2)
`session_id`	Unique Session Identifier passed on the launch URL (AICC_SID value from the launch url)
`AU_password`	Assignable Unit (content object) specific password. This parameter is optional.
`AICC_Data`	Data specific to the command. If the content was sending information to the LMS, this is the parameter under which the data would be sent. The AICC_Data value must be URL encoded.

Usage Rules

- The `AICC_Data` value is URL-encoded.
- The name/value pairs can appear in any order.
- If an optional value is to be omitted, the name must also be omitted.
- The name of each parameter is not case sensitive.

Most Common Commands

- `PutParam` sends data to SAP SuccessFactors.
- `GetParam` retrieves data from SAP SuccessFactors.

Tips

- `%0D` is the hex value for carriage return.
- `%0A` is the hex value for line feed.
- `%3D` is the hex value for equals (=).
- `%26` is the hex value for an ampersand (&).

4.4 Standard AICC_DATA Values

This section provides information about standard AICC_Data values.

An AICC post is comprised of a series of name/value pairs (see the Required Data Format page). The `AICC_Data` parameter is the name/value pair that contains the specific data being sent from the content to the LMS. The

`AICC_Data` parameter is also comprised of name/value pairs which define the information being sent. The `AICC_Data` value is URL-encoded.

Groups and Parameters

Group	Parameters
[Core] (required)	`Lesson_status`
	`Lesson_location`
	`Score`
	`Time`
[Objectives_Status]	`J_ID.1`
	`J_Score.1`
	`J_Status.1`
[Core_Lesson]	Data is undefined and may be unique to each lesson

Usage

> **Note**
>
> A carriage return and line feed must be included after each name/value pair and group tag (i.e., [core]).

- The `AICC_Data` value is URL-encoded.
- The Name/value pairs can appear in any order.
- If an optional value is to be omitted, the name must also be omitted.
- The name of each parameter is not case sensitive.
- The [core] group and all of its parameters are required.

Example

> **Sample Code**

```
[core]
lesson_location=end
lesson_status=pass
score=87
time=00:23:15
[core_lesson]...
```

URL-encoded, the string:

Sample Code

```
[core]%0D%0Alesson_location%3Dend%0D%0Alesson_status
%3Dpass%0D%0Ascore%3D87%0D%0Atime%3D00:23:15%0D%0A
[core_lesson]%0D%0AThis%20is%20sample%20text%20for
%20the%20core_lesson%20parameter
```

The full AICC post would look like this:

Sample Code

```
https://customer.lms.com/learning/PwsAicc?
Command=PutParam&Version=2.2&Session_id=C1234
&aicc_data=[core]%0D%0Alesson_location%3Dend%0D%0A
lesson_status%3Dpass%0D%0Ascore%3D87%0D%0Atime%3D00:23:15%0D%0A
[core_lesson]%0D%0AThis%20is%20sample%20text%20for%20the%20
core_lesson%20parameter
```

AICC [Core] Group Overview [page 58]
The Core group contains the main data elements for your content. The Core group and all of its parameters are required when sending an AICC post.

AICC [Core_Vendor] [page 62]
The `Core_Vendor` group is used to pass data to the lesson on launch.

AICC [Core_Lesson] [page 63]
This section provides information about using `AICC [Core_Lesson]`.

Example of AICC Supported Data Elements [page 63]
This example shows SAP SuccessFactors Learning AICC supported data elements.

4.4.1 AICC [Core] Group Overview

The Core group contains the main data elements for your content. The Core group and all of its parameters are required when sending an AICC post.

Parameters	Description
`Lesson_status`	Used for passing completion status to SAP SuccessFactors Administration
`Lesson_location`	Used to store bookmarking data
`Score`	Score (one per content object)
`Time`	HHHH:MM:SS.SS (accumulative value - adds to previous time in the LMS

AICC Parameter Lesson_status and Lesson_status flag [page 59]
`Lesson_status` and `Lesson_status` flag determine the completion status of the content module.

AICC Parameter Lesson_Location [page 60]
Bookmarking information can be stored using the AICC `Lesson_Location` value.

AICC Parameter Score [page 61]
`Score` is an indication of the performance of the student during his/her last session in the content object.

AICC Parameter Time [page 62]
`Time` is the accumulated time of all the student sessions in the content.

Parent topic: Standard AICC_DATA Values [page 56]

Related Information

AICC [Core_Vendor] [page 62]
AICC [Core_Lesson] [page 63]
Example of AICC Supported Data Elements [page 63]

4.4.1.1 AICC Parameter `Lesson_status` and `Lesson_status` flag

`Lesson_status` and `Lesson_status` flag determine the completion status of the content module.

Usage Notes

- The first letter of the `Lesson_status` value is processed, so `P`, `pass`, and `passed` are treated the same.
- `Lesson_status` and `Lesson_status` flag are separated by a comma (for example: `P,L`).
- The flag information is **not** passed back as part of the `Getparam` response.

`Lesson_status`

`Lesson_status` Value	Meaning
`Passed` or `p`	Necessary number of objectives in the lesson were mastered, or the necessary score was achieved. Student is considered to have completed the lesson and passed.
`Completed` or `c`	The lesson may or may not be passed, but all the elements in the lesson were experienced by the student. The student is considered to have completed the lesson.

`Lesson_status` Value	Meaning
`Failed` or `f`	The lesson was not passed. All the lesson elements may or may not have been completed by the student.
`Incomplete` or `i`	The lesson was begun but not finished.
`Not attempted` or`n` or `na`.	This is the initial `Lesson_status` value set by the LMS. Not attempted means that the student did not even begin the lesson.

`Lesson_status` flag

Flag Value	Meaning
`Time_out` or `T`	This indicates the lesson ended because the lesson has determined an excessive amount of time has elapsed.
`Suspend` or `S`	This indicates the student leaves the lesson with the intent of returning to it later at the point where he/she left off. The LMS will send a `resume` `Lesson_status` flag the next time the content is launched.
`Logout` or `L`	This indicates that the student logged out of the LMS from within the lesson instead of returning to the LMS to log out. The intent is to require re-authentication from the student.

Parent topic: AICC [Core] Group Overview [page 58]

Related Information

AICC Parameter Lesson_Location [page 60]
AICC Parameter Score [page 61]
AICC Parameter Time [page 62]

4.4.1.2 AICC Parameter `Lesson_Location`

Bookmarking information can be stored using the AICC `Lesson_Location` value.

Each new `Lesson_location` post overwrites the existing value. The `Lesson_Location` field is limited to 255 characters.

Parent topic: AICC [Core] Group Overview [page 58]

Related Information

AICC Parameter Lesson_status and Lesson_status flag [page 59]
AICC Parameter Score [page 61]
AICC Parameter Time [page 62]

4.4.1.3 AICC Parameter `Score`

`Score` is an indication of the performance of the student during his/her last session in the content object.

This score may be determined and calculated in any manner that makes sense to the program designer. For instance, it could reflect the percentage of objectives complete, it could be the raw score on a multiple choice test, or it could indicate the number of correct first responses to the embedded questions in the content object.

The score can be a stand-alone value, or may be followed by two values: a maximum and minimum. Where the maximum is the largest score the student could have achieved with the interactions experienced and the minimum is the smallest score that the student could have achieved with the interactions experienced.

The format of `Score` is a decimal number or blank for all three values. If three values are passed, they are separated by commas, and the order is significant: Score, Maximum, Minimum

Sample Code

```
SCORE = 79
```

The score typically represents a percentage with a minimum and maximum. This is a raw score of 8 with a maximum of 10 and a minimum of zero. As a percentage: 80%.

Sample Code

```
Score = 8.0, 10.0, 0.0
```

Parent topic: AICC [Core] Group Overview [page 58]

Related Information

AICC Parameter Lesson_status and Lesson_status flag [page 59]
AICC Parameter Lesson_Location [page 60]
AICC Parameter Time [page 62]

4.4.1.4 AICC Parameter `Time`

`Time` is the accumulated time of all the student sessions in the content.

The format of `Time` is `HHHH:MM:SS.SS`. It's an integer number representing hours, followed by a colon, an integer from 00 to 59 representing minutes, followed by a colon and a decimal or integer from 00 to 59.99 representing seconds. Three numbers, separated by colons, are always required, even if only seconds or minutes are represented.

Sample Code

```
TIME=00:29:00
```

Note

SAP SuccessFactors Learning only records the time value from the last AICC post sent prior to `ExitAu` being called. This value is added to the total time stored in the LMS from previous content sessions. If the `ExitAu` post isn't received by SAP SuccessFactors Learning to indicate the end of the content session, the learner's time won't be recorded.

Parent topic: AICC [Core] Group Overview [page 58]

Related Information

AICC Parameter Lesson_status and Lesson_status flag [page 59]
AICC Parameter Lesson_Location [page 60]
AICC Parameter Score [page 61]

4.4.2 AICC [Core_Vendor]

The `Core_Vendor` group is used to pass data to the lesson on launch.

`Core_Vendor` data is only passed from the LMS to the content, and cannot be reset by the content. This data may be set automatically by the AICC import assistant from the `Core_Vendor` parameter of the AU import file. This data group is limited to 4096 characters.

Parent topic: Standard AICC_DATA Values [page 56]

Related Information

AICC [Core] Group Overview [page 58]

AICC [Core_Lesson] [page 63]
Example of AICC Supported Data Elements [page 63]

4.4.3 AICC [Core_Lesson]

This section provides information about using `AICC [Core_Lesson]`.

The `Core_Lesson` group is contained in the `AICC_Data` section of a `PutParam` command. `Core_Lesson` contains data that is created by the content and stored by the LMS to be passed back to the content the next time it is run. Typically this data is used to store learner progress data.

Any new `Core_Lesson` data overwrites the existing data. `Core_Lesson` is limited to 4096 characters.

Parent topic: Standard AICC_DATA Values [page 56]

Related Information

AICC [Core] Group Overview [page 58]
AICC [Core_Vendor] [page 62]
Example of AICC Supported Data Elements [page 63]

4.4.4 Example of AICC Supported Data Elements

This example shows SAP SuccessFactors Learning AICC supported data elements.

Sample Code

```
[CORE]
Lesson_Status
Lesson_Location
Score (raw, min, max)
Time
Exit Flag
[CORE_LESSON]
[OBJECTIVES_STATUS]
Objective.n.status
Objective.n.score (raw,min,max)
Objective.n.id
[CORE_VENDOR]
```

Parent topic: Standard AICC_DATA Values [page 56]

Related Information

AICC [Core] Group Overview [page 58]
AICC [Core_Vendor] [page 62]
AICC [Core_Lesson] [page 63]

4.5 AICC `Max_Normal`

The AICC `Max_Normal` value is the maximum number of assignable units (content objects) that may be taken for credit simultaneously.

AICC Max Normal represents the maximum number of content objects that a user can access without completing it and receiving credit for it. If a user accesses a subsequent time that exceeds that number then the content object will not grant credit. When the number of incomplete content objects exceeds the Max Normal value, subsequent launches of AICC content are set with `credit="no credit"`. AICC `Lesson_Status` and `Score` values cannot be updated.

For example, if an item has five content objects and the max normal value for the item is set to three, then a learner can access up to three content objects without completion. If they access and don't complete a fourth content object, any AICC content object they access is launched with `Credit="No Credit"`, and the `Lesson_Status` and `Score` values are locked.

The completion value used for the max normal calculation is a SAP SuccessFactors calculation, not an AICC calculation. As a result, the Max Normal calculation is based on all content objects in an item whether they're AICC content objects or not (this also includes exam objects). However, only AICC (and SCORM) content objects are affected if the Max Normal value is exceeded.

We recommend that you set AICC Max Normal to 99.

4.5.1 Setting AICC Max Normal Default

Set AICC Max Normal value to set a default control how many content objects can be open for credit at one time in a learning item.

Context

AICC Max Normal represents the maximum number of content objects that a user can access without completing it and receiving credit for it. If a user accesses a subsequent time that exceeds that number then the content object will not grant credit. When the number of incomplete content objects exceeds the Max Normal value, subsequent launches of AICC content are set with `credit="no credit"`. AICC `Lesson_Status` and `Score` values cannot be updated.

New courses (learning items) get the AICC max normal value from *Global Application Settings* by default when you create new learning items. Individual courses can override the default. Imported AICC courses can set their max normal values in the CRS definition.

Procedure

1. Go to SAP SuccessFactors Learning Administration and then go to *System Administration* > *Configuration* > *Global Applications Settings*.
2. Set the default value in *AICC Max-Normal*.
3. Click *Apply Changes*.

4.5.2 Setting AICC Max Normal In a Course

Set AICC `Max_Normal` to set a control on a learning item for how many content objects can be open for credit at one time in a learning item.

Context

> **Note**
>
> It is unusual to set `Max_Normal` value in an individual course. In most cases, you inherit the default value, which we recommend that you set to **`99`**.

AICC Max Normal represents the maximum number of content objects that a user can access without completing it and receiving credit for it. If a user accesses a subsequent time that exceeds that number then the content object will not grant credit. When the number of incomplete content objects exceeds the Max Normal value, subsequent launches of AICC content are set with `credit="no credit"`. AICC `Lesson_Status` and `Score` values cannot be updated.

New courses (learning items) get the AICC max normal value from *Global Application Settings* by default when you create new learning items. Individual courses can override the default. Imported AICC courses can set their max normal values in the CRS definition.

Procedure

1. Go to SAP SuccessFactors Learning Administration and then go to *Learning Activities* > *Items*.
2. Open the course (learning item) that you want to edit.
3. Go to *Online Content* > ⚙ > *(Settings)* > *Completion* (tab).

4. Set the value in *AICC Max-Normal*.
5. Choose *OK* to save your changes.

4.5.3 AICC `Max_Normal` Inheritance

You set the default AICC `Max_Normal` in *Global Applications Settings*, but you can also override the default on the course (learning item) level.

You set the default `Max_Normal` in *System Administration* > *Configuration* > *Global Applications Settings* > *AICC Max-Normal*, but you can override it in individual courses. The override can be populated in one of these places:

- You can set the value of `Max_Normal` in *Learning Activities* > *Items* > *Online Content* > *Settings* > *AICC Max-Normal*.
- When you import an AICC course in *System Administration* > *Tools* > *AICC Import*, you can set `Max_Normal` in the CRS import file.

4.6 Supported AICC Import and Export Files

SAP SuccessFactors Learning supports these AICC files for import and export.

AICC defines a method of importing content definitions into a Learning Management System (LMS). Importing the same files twice will produce two sets of data, so any edits that need to be made after importing the content definitions should be made to the actual records in SAP SuccessFactors.

> **Note**
>
> When you import AICC files to SAP SuccessFactors Learning, you do **not** import the actual content. Instead, you import the definition (or setup) of the content.

- The Course file (*.crs): This file contains course-level information about the course as a whole.
- The Assignable Unit file (*.au): This file contains technical information relating to all the assignable units (content objects) in the course (i.e. content object launch url).
- The Descriptor file (*.des): This file contains descriptive information relating to every course element in the course (i.e. content object description). It is used as the basic cross-reference file showing the correspondence of system generated IDs with user-defined IDs for every element.
- The Course Structure file (*.cst): This file contains the basic data on the content structure of the course. It includes all of the assignable units (content objects) and blocks (folders) in the course.

4.7 When and How to Use the AICC Document Wrapper

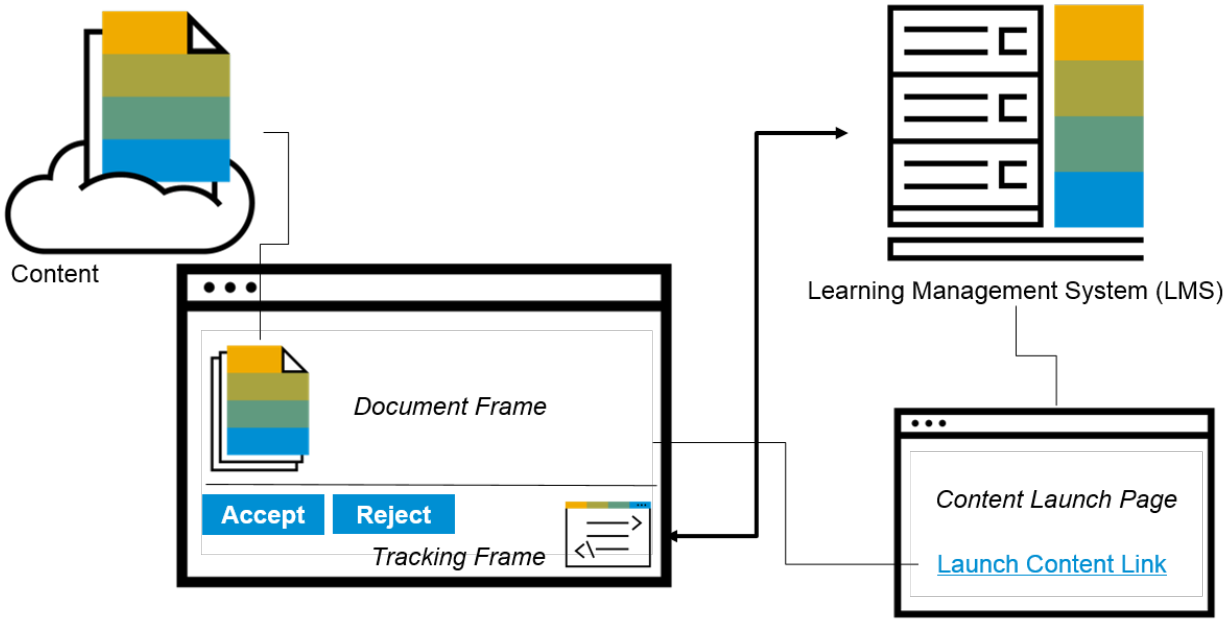
The AICC document wrapper wraps online documents or other browser-based content so that SAP SuccessFactors Learning can record both the completion of the content and the time that the learners spent in the content.

Use the AICC wrapper when content doesn't contain any mechanism for communicating back to SAP SuccessFactors Learning. Don't use it if the online content already has a mechanism to communicate status back. In other words, use it for documents like PDF files and not for active content like SCORM content.

When the content object launches with an AICC wrapper, the learners' browsers call for the appropriate document or object to be loaded into the wrapper frame. The tracking frame of the wrapper then communicates data to the LMS using HTTPS for completion. Objects to be loaded into the wrapper can live at any location the user can access in a browser.

There are two types of wrappers: internal and external:

- The internal version is simpler to set up and requires only that the document is available on a URL. You don't need a tracking content server. It supports only one confirmation text set per language we support.
- The external version is more complex to set up but is more useful for companies that deploy our cross-domain solution to create a tracking content server. As part of setting up the tracking content server, you extract wrapper.html. Content objects launch wrapper.html and pass the document as a parameter. Because you have full control over wrapper.html, you can clone it to create many wrappers, each with a different confirmation text and in any language.



Architecture of the AICC Wrapper

Using the External AICC Wrapper to Create Confirmation Statements [page 68]

Use the external AICC Wrapper to create confirmation statements if you manage your content on a separate content server and you want users to confirm that they have read documents (and you want to store their confirmation).

Creating Content That Users Can Confirm Reading [page 69]

Use the internal AICC wrapper to set up content that users read and then agree or disagree to (or confirm that they have read). Users' decisions are recorded as a learning event.

Changing Internal AICC Wrapper Text Confirmation Text [page 70]
Change the internal AICC wrapper text to change what users read when they are presented with a document and asked to confirm that they have read the document.

Setting AICC Wrapper Confirmation Text [page 71]
Set the AICC wrapper confirmation text that you want to appear to learners as the buttons and text to agree with content or disagree with content.

4.7.1 Using the External AICC Wrapper to Create Confirmation Statements

Use the external AICC Wrapper to create confirmation statements if you manage your content on a separate content server and you want users to confirm that they have read documents (and you want to store their confirmation).

Prerequisites

Extract wrapper.html to your content server. You typically extract the file as part of your cross-domain solution when you first set it up. You need to know the location of wrapper.html as part of this process.

Context

You often want to create content that users agree to or disagree to. For example, if you have a Standard Operating Procedure (SOP) and you need learners to agree or disagree to the SOP, you can wrap it in an external AICC wrapper and record that users have at least agreed to reading it. Use the external AICC wrapper when you manage your content on a content server.

> **Note**
>
> - Don't select the *Use AICC Wrapper* checkbox when you're using an **external** AICC wrapper.
> - With an external AICC wrapper, the content is always launched in a new browser tab.

Procedure

1. Add the document that users agree to on a URL on your content server.
2. Go to SAP SuccessFactors Learning Administration and then go to *Content* > *Content Objects*.
3. Click *Add New* to add a new content object.

4. Complete the information in the *Summary* section.
5. In *Edit the Launch Method for the Content Object*, select *AICC*.
6. In *Filename*, type the relative URL to the wrapper.html file that you extracted as part of your cross-domain solution.

   For example, customers typically extract to `/content/wrapper/wrapper.html`.
7. In *Parameters*, type **`docURL=<URL to your document>`**

   For example, `docURL=https://content.server.com/docs/my_file.pdf`
8. Click *Apply Changes*.

Task overview: When and How to Use the AICC Document Wrapper [page 67]

Related Information

Creating Content That Users Can Confirm Reading [page 69]
Changing Internal AICC Wrapper Text Confirmation Text [page 70]
Setting AICC Wrapper Confirmation Text [page 71]

4.7.2 Creating Content That Users Can Confirm Reading

Use the internal AICC wrapper to set up content that users read and then agree or disagree to (or confirm that they have read). Users' decisions are recorded as a learning event.

Prerequisites

Before you set up content that users can agree or disagree to (and record their responses), configure the boilerplate text. You can set up the boilerplate text and translate it to different languages in *System Administration* > *Configuration* > *Global Applications Settings* > *AICC Wrapper Settings*.

Context

You often want to create content that users agree to or disagree to. For example, if you have a Standard Operating Procedure (SOP) and you need learners to agree or disagree to the SOP, you can wrap it in an internal AICC wrapper and record that users have at least agreed to reading it.

> **Note**
>
> This method describes using the internal AICC Wrapper file, which most customers use. If you manage your content externally, you can also use the external AICC Wrapper file.

Procedure

1. Add the document that users agree to on a URL that is accessible to your learners.
2. Go to SAP SuccessFactors Learning Administration and then go to *Content* > *Content Objects*.
3. Click *Add New* to add a new content object.
4. Complete the information in the *Summary* section.
5. In *Edit the Launch Method for the Content Object*, select *AICC*.
6. Select *Use AICC Wrapper*.
7. In *Filename*, paste the URL where the document is stored.

   For example, if you want users to confirm that they have read my_file.pdf on your intranet, the URL in *Filename* could be `HTTP://intranet.company.com/docs/my_file.pdf`.
8. Click *Apply Changes*.

Task overview: When and How to Use the AICC Document Wrapper [page 67]

Related Information

Using the External AICC Wrapper to Create Confirmation Statements [page 68]
Changing Internal AICC Wrapper Text Confirmation Text [page 70]
Setting AICC Wrapper Confirmation Text [page 71]

4.7.3 Changing Internal AICC Wrapper Text Confirmation Text

Change the internal AICC wrapper text to change what users read when they are presented with a document and asked to confirm that they have read the document.

Context

You often want users to confirm that they have read a document. For example, you can show them a Standard Operating Procedure (SOP) as part of their training and ask that they confirm that they have read it. You can wrap the document in the AICC wrapper to record their response. By default, the text is: I confirm that I have read and understand all the material contained in this document in English. The confirmation buttons are Agree and Disagree in English. If you want to change the text, you can do it in the global variables.

Procedure

1. Go to SAP SuccessFactors Learning Administration and then go to *System Administration* > *Configuration*.
2. In *AICC Wrapper Text*, type the new instructions to users in *Confirmation Text*, the new affirmation button text in *Agree Button Text*, and the new decline text in *Disagree Button Text*.
3. Translate the text by clicking the localize button ().

Task overview: When and How to Use the AICC Document Wrapper [page 67]

Related Information

Using the External AICC Wrapper to Create Confirmation Statements [page 68]
Creating Content That Users Can Confirm Reading [page 69]
Setting AICC Wrapper Confirmation Text [page 71]

4.7.4 Setting AICC Wrapper Confirmation Text

Set the AICC wrapper confirmation text that you want to appear to learners as the buttons and text to agree with content or disagree with content.

Context

Some online content uses an AICC wrapper. An AICC wrapper presents the user with some text and then asks the user to agree or disagree with the content object. For example, you can wrap a Standard Operating Procedure (SOP). Users read the SOP and then see some text that says that they confirm that they've read and understood it. They then click an agreement or a disagreement button. The settings control the text that users see.

Procedure

1. Go to Learning Administration, and then go to *System Administration* > *Configuration* > *Global Applications Settings*.
2. In the page, find *AICC Wrapper Settings*.
3. Type new values for the confirmation prompt, and the agree and disagree buttons.

Choice	Description
Confirmation Text	The text that explains their task. Do they agree that they have read and understood the material, for example, or do they not agree?
Agree Button Text	The text that appears on the affirmative button: the button that users click when they agree that they have read and understood the material.
Disagree Button Text	The text that appears on the dissension button: the button that users click when they disagree that they have read and understood the material.

4. To translate any of the values click .

   When you click , you can translate your text into multiple locales.
5. Click *Apply Changes*.

Task overview: When and How to Use the AICC Document Wrapper [page 67]

Related Information

Using the External AICC Wrapper to Create Confirmation Statements [page 68]
Creating Content That Users Can Confirm Reading [page 69]
Changing Internal AICC Wrapper Text Confirmation Text [page 70]

5 Structured Content Troubleshooting

Although we don't officially support third-party content or third-party content servers, we do offer information about common problems and diagnostic tools.

SAP SuccessFactors offers Premium Content Management (PCM), which is available under a separate license. With Premium Content Management, content integration specialists deliver expertise in AICC and SCORM eLearning standards, common course development tools, the SAP SuccessFactors Learning application and mobile app, and the SAP SuccessFactors content hosting platform. PCM experts help to catch potential mistakes before they become issues, assist with long-term management tasks, and anticipate the potential impact of individual course issues on the customer's larger library of courses. We recommend Premium Content Management to address common pain points in structured content like AICC or SCORM. Please contact SAP SuccessFactors sales representatives for more information about Premium Content Management.

AICC content can be difficult to troubleshoot, particularly its communication from its content server back to the Learning Management System (LMS). If you decide not to use Premium Content Management from SAP SuccessFactors, you troubleshoot your content with options ranging from third-party tools to built-in utilities, so you need to understand AICC, internet communication protocols, and cross-domain content delivery. Some companies do elect to manage their own content, so we offer some utilities to help and some advice for third-party tools.

5.1 Checking the Launch Method of Online Content

When the launch method is set incorrectly, you can see communication problems between the content and the Learning Management System (LMS).

Context

When troubleshooting content, the launch method is the easiest thing to check. If the launch method is set incorrectly, tracking and communication won't work properly.

Procedure

1. Go to SAP SuccessFactors Learning Administration and then to go *Content* > *Content Objects*.
2. Find and open the content object that is having trouble.
3. Go to *Launch Method*.
4. Check the launch method.

   For example, if the content is SCORM 1.2, make sure the content is specified SCORM 1.2.

5. Check *Content Parameters* to make sure that they are configured correctly.

5.2 Checking to See if Online Content is Communicating Status About a User

Check to see if structured content is communicating with a user if you are unsure that course content is communicating completion status back to the Learning Management System (LMS).

Context

When you check to see if online content is communicating status about a user to the LMS, you are running the simplest communication test. We recommend that you try the simple test first because it is easy to set up, but it won't show you everything that the LMS stores. After you use the simple method, we recommend the utility pages to check the communication more fully.

Procedure

1. Go to SAP SuccessFactors Learning Administration and then go to *People* > *Users*.
2. Choose *Add New* to add a test user, and then assign structured content to the user in *Assigned Items*.
3. Log in as the user and begin the content, but do not complete it.

   > **Note**
   >
   > There is a chance that the course content and the LMS are communicating but that the LMS isn't receiving the completion. For this reason, troubleshoot the content to look for the completion marker.

   If it is working properly, you should see the content marked as complete in the content structure page or learning history.
4. After you spend some time in the content, open the test user in SAP SuccessFactors Learning Administration, *People* > *Users*.
5. Go to the *Completed Items* tab, find the corresponding item, and then choose *View Details*.
6. Under *Learning History Details*, look for the status reported back to the LMS.

   > **Note**
   >
   > The status data can differ by content type, but you should see the initial access, the last access, and so on.

If you see status that resembles your test, then completion status is reported back to the LMS. If you are investigating a single user whose status is not reported, then there could be a problem in the learner's profile.

If you **don't** see status that resembles your test, then chances are that it is not a user problem but instead a communication problem. You can use the test page utilities and third party communication tools to investigate the issue further.

5.3 Using Fiddler to Watch Traffic Between Course Content and SAP SuccessFactors Learning

Use Fiddler to watch traffic between course content and SAP SuccessFactors Learning if you think the content isn't communicating status correctly back to the Learning Management System (LMS) and you want to look at the communication for trouble.

Procedure

1. Download and install Fiddler.
2. Go to options and then select *Capture HTTPS CONNECTs* and *decrypt HTTPS traffic*.
3. Connect to your content or use test pages to generate traffic.
4. In the log, look for `/learning/PwsAicc` for AICC communication and `/learning/ScormRteServlet` for SCORM communication.

5.4 Communication Between SCORM 1.2 Learning Content and SAP SuccessFactors Learning

The SCORM 1.2 follows a standard API communication process.

When SCORM content is launched, the first thing it does is to search for the SAP SuccessFactors Learning API. It looks in all the frames in its own window first. If it does not find the API object, it attempts to search in the window that launched it. In SAP SuccessFactors Learning, the API is hosted in the learner application, so the content will have to search the opener window to find the API. There are several reasons why the content may not be able to find the API.

If the content is hosted under a separate DNS name (server name) from the LMS, the content will not be able to access the API. The browser, for security reasons, will prevent the content from searching in the opener window and will usually give an "Access Denied" error. For more information, please see Cross-Domain Communication Restriction.

The code that looks for the API may need to be altered. Typically there is an `APIWrapper.js` file that contains functions provided by ADL (writers of the SCORM spec). The functions that locate the API are `getAPI` and `findAPI`. The `getAPI` function calls the `findAPI` function and specifies a window to search. The `getAPI` call to search the opener window is usually:

Sample Code

```
theAPI = findAPI(window.opener);
```

If the SCORM content is using frames, this function call does not make sense to the browser because it looks for the opener of a particular frame. You might to fix this line of code in your SCORM package to search the opener of the frame set:

Sample Code

```
theAPI = findAPI(top.window.opener);
```

If the content is using an opener window to format the content's browser window, the content has two levels out from the LMS so the correct call is also:

Sample Code

```
theAPI = findAPI(top.opener.opener);
```

5.5 Common Problems and Solutions for SCORM 1.2 and AICC Content

When you have communication problems with AICC content while using SCORM 1.2, check for common issues.

Learners See API Not Found When They Launch a Course

To diagnose the problem, check the trace log to verify that there's a `learning/PwsAicc` URL.

- If you see `learning/PwsAicc`, verify the SCORM values.
- If you **don't** see `learning/PwsAicc`, you didn't get a SCORM connection, which caused the *API Not Found* error. There are several possible reasons for this:
  - If the issue only happens with Microsoft Edge, the `https://*.plateau.com` domain could be missing from the *Trusted Sites* list. Follow Microsoft's instructions to investigate your trusted sites and add the domain if it's missing. If the issue is happening in a single region, it's possible that all users of this region don't have this listed as a trusted site. The regional variation is common because different Microsoft Edge settings can be pushed to different regions with the Windows Group Policy.
  - The course might not be SCORM 1.2 compliant. Test the course on a different LMS to see if you can replicate the error. Use CloudScorm LMS at https://cloud.scorm.com. Register to use the site, import the content, launch, and then collect and check the logs.
  - The cross-domain solution (Proxlet) might not be set up correctly. This is only applicable if you're using SCORM 1.2 and the content **isn't** hosted on SAP SuccessFactors Content Hosting. Go to *Content Object* > *Cross Domain* and check that all URLs are listed correctly.

- There may be a problem with your SAP SuccessFactors Content Hosting server. If you recently implemented SAP SuccessFactors Content Hosting, and you recently found the error, go to the system diagnostics to check all services. If they aren't running properly and you're an SAP SuccessFactors Content Hosting customer, open a Customer Support ticket.

Learners are Logged out During a Session

If learners are logged out unexpectedly, identify the `KeepAlive` session ping, which keeps the session active. Every 5 minutes , the Online Structure page sends a `KeepAlive` ping to SAP SuccessFactors Learning so that it renews the session timeout. If the ping is operating correctly, the session shouldn't expire when the course is open.

> **Note**
>
> You set the ping interval in *System Admin* > *Configuration* > *System Configuration* > *LMS_ADMIN* > *contentStructurePageKeepAliveInternal*.

Check the trace log to verify that there's a `checkStudentStatusAction.do` URL.

- If you see `checkStudentStatusAction.do`, then timeout isn't the problem. It also indicates that the student hasn't navigated away from the *Online Structure* page.
- If you don't see `checkStudentStatusAction.do` and the learners spent **less** than 5 minutes in the course, then timeout isn't the problem because the learners didn't stay long enough for a `KeepAlive` ping.
  If you don't see `checkStudentStatusAction.do` and the learner spent **more** than 5 minutes in the course, then the learner either closed SAP SuccessFactors Learning or navigated away from the *Online Structure* page. In this case, the SCORM connection is deactivated and the course can't send the completed information to SAP SuccessFactors Learning.

Learners Complete Assessments but the Assessments Don't Go to Learning History

Check the trace log to verify that there's a `LESSON_STATUS` value of *Completed* or *Passed*.

- If the status is *Completed* but the item didn't go into the Learning History, you need to make sure that the content object triggers the item completion. Go to *Learning Activities* > *Items* > *Online Content* > *(Settings)* and ensure the *Add to History on Completion of All Content* selection is set to *ON*. This configuration is applied to the item when it's assigned, so any item that is already assigned must be reassigned to get the new settings.
- If the status is anything other than *Completed* or *Passed* (for example, *Incomplete*, *Failed*, or empty), the issue is coming from the content itself. Contact the vendor to explain the problem.

5.6 Content Object Launch Method Fields in SAP SuccessFactors Learning

A content object's launch method defines how SAP SuccessFactors Learning should start the online content.

To specify the content player and command line parameters it needs to launch the content object, go to *Content* > *Content Objects* > *Launch Method*.

This field...	Is used as follows...
Content Player	Select this button if you want the player for this content to be selected based on your explicit choice of a player application file located in one specific place on your network. See also, Content Players.
Document Type	Select this button if you want the operating system to determine the player for this content, based upon the file types and associations that are registered with the operating system.
AICC	Select this button if this courseware is an AICC assumable unit. By doing so, you establish the proper connection between SAP SuccessFactors Learning and the AICC courseware. All AICC content objects should be web browser-based. This type of content will always launch in a new window.
Browser	Select this button to have the system use the user's browser to launch and run the selected content object. If the content object is opened in the same browser window, the URL must be secure using HTTPS. This type of content will always launch in a new window.
SCORM 1.2	Select this button if the courseware is SCORM-based and version 1.2. This establishes the proper connection between SAP SuccessFactors Learning and the SCORM courseware.
SCORM 2004	Select this button if the courseware is SCORM-based and version 2004, 2nd edition. This establishes the proper connection between SAP SuccessFactors Learning and the SCORM courseware. **Note** While SCORM 2004 3rd edition isn't a supported version, this setting can be used with SCORM-based, version 2004, 3rd edition courseware if the features match those available in the 2nd edition. Features unique to SCORM 2004 3rd edition, such as branching, won't work.

This field...	Is used as follows...
SCORM 2004 4th Edition	Select this button if the courseware is SCORM-based and version 2004, 4th edition. This establishes the proper connection between SAP SuccessFactors Learning and the SCORM courseware.
Launch in a New Browser Window	You can set course content to launch in a new browser window. (For the *Launch in a New Browser Window* option to be available, ensure that you've selected the *Display "Launch in a new Browser Window" option for content objects* option in *System Administration* > *Configuration* > *Global Applications Settings*.) **Note** Administrators can configure `overrideContentLaunchInNewWindow=true` in *System Administration* > *Configuration* > *System Configuration* > *LMS_ADMIN* to override the content-object level launch method for all content **except** browser-based and external content.
Content Player	The player you choose is always used for the particular content associated with this content object. This field is only used when the *Content Player*
Use AICC Wrapper	The most common reason to use the AICC wrapper is for tracking the completion of a document. For example, if your training consists of reading a Standard Operating Procedure (SOP), you can upload the SOP as AICC (Aviation Industry Computer based training Committee) standard wrapper to track whether users verified the SOP are read or did not. The process for setting up the AICC wrapper system is typically handled during initial implementation. **Restriction** AICC wrapper is **not** supported for offline player, but is supported for mobile devices.

This field...	Is used as follows...
Content URL	When using a Content Player, type the URL of the actual content. If your content player is set up properly, it knows which executable file to use to launch that content file. When using *Document Type* or an AICC course, type the URL of the file or AICC course in the *Filename* field. If any parameters need to be passed to the executable, you can enter them into the *Parameters* field. **Note** A content launch error will occur if any unsupported characters are used in the launch URL or any file that the content launches. Because the launch URL is created based on the imported content zip file, please ensure all files in the import zip file and any content package IDs contain only supported characters. For more information, see Characters Not Supported in Content File Names [page 88].
Offline Filename	The location of the content object in the content archive of a downloadable content object. For example, if the content object is in the archive at the <archive root>/resources/Lesson_1.htm, the value is resources/Lesson_1.htm. If the Offline Filename value doesn't match the actual location in the archive, Learning Offline can't launch the object. If you import a SCORM object, the value is pre-populated with the default SCORM value.
Parameters	Use this field to provide the necessary parameters that need to be passed when launching the course.

This field...	Is used as follows...
Folder Access Level	User this field to set specific access folder restrictions for iContent. The value provided in the field must start with a forward slash (/). For example, if you set a value of **`/self-managed/java101`**, the content will only launch if the file is within or below the **`java101`** folder. **Note** This field is controlled by the `enableiContentSpecificAccessFolderRestriction` setting (*System Administration* > *Configuration* > *System Configuration* > *WEB_SECURITY*). By default, this setting is `false` and the field is hidden. If `enableiContentSpecificAccessFolderRestriction=true`, administrators can use the `iContentGeneralAccessFolders` setting to define any files within iContent folders intended for general access. The value for this setting is empty by default. To define multiple folder paths, separate each complete folder path with a comma.
Enable Mobile Access	Select this checkbox to identify content that launches well in mobile devices, as well as in a browser. If you select Enable Mobile Access for a particular content and enabled the Mobile Readiness Indicator during implementation, then the content displays a mobile readiness indicator. **Note** Test your content on mobile devices before selecting this checkbox in production.

Related Information

SAP SuccessFactors Learning Course Home
Settings That Control Online Content in a Learning Item

5.7 Potential Integration Issues

Before you import structured content to SAP SuccessFactors Learning, check for potential integration issues.

Content Import

Standard	Potential Issue	Notes
SCORM 1.2 SCORM 2004	Metadata Schema Requirement	SAP SuccessFactors Learning requires the `imsmanifest` to contain at least one `metadata` tag containing the adl schemaversion. **Sample Code** `<metadata> <schema>ADL SCORM</schema> <schemaversion>1.2</schemaversion> </metadata>`
AICC	AU `Command_Line` Parameter	The AU import file shouldn't contain any `command_line` parameter values. SAP SuccessFactors Learning doesn't generate a valid AICC object for any assignable unit with a `command_line` value.
AICC	ORT file	The SAP SuccessFactors Learning AICC import assistant doesn't support ORT files.

Content Launch and Initialization

Standard	Potential Issue	Notes
SCORM 1.2	Finding the SCORM API	SAP SuccessFactors Learning opens content into its own window so the content must search under the opener window to locate the SCORM API.

Standard	Potential Issue	Notes
AICC SCORM 1.2	Content Window size and format	SAP SuccessFactors Learning doesn't allow a user to format the content window size or appearance. AICC content is launched via a standard hyperlink. SCORM 1.2 content is launched from SAP SuccessFactors Learning using a `window.open` function call with the following settings (`toolbar=1,location=0,directories=0,status=1,menubar=1,scrollbars=1,resizable=1`)

Content Communication

Standard	Potential Issue	Notes
AICC SCORM 1.2	Exit Flag (`cmi.core.exit`)	SAP SuccessFactors Learning exits a learner from the SAP SuccessFactors Learning application when an exit flag value of `logout` is received.
AICC SCORM 1.2	Objective Data Tracking	AICC and SCORM Objective data isn't tracked unless the objectives exist in the SAP SuccessFactors Learning database and are associated with the content module. The objective IDs for all incoming AICC or SCORM objective data must match existing objectives in the SAP SuccessFactors Learning database. SAP SuccessFactors Learning SCORM import assistant generates and associates objectives from the `imsmanifest` metadata, but the AICC import assistant does `not` generate objective records.

5.8 Structured Content APIs Depend on Proper Usage of Session

The structured content API system depends on users closing sessions to record some values.

When you troubleshoot structured content API systems, always verify that users are using the standard navigation buttons instead of using the browser buttons. Individual API calls are attached to the navigation buttons, and when users don't use the buttons, the API calls are never made.

For example, if users close the browser window with the window close button (standard Microsoft Windows functionality), the content may not make the appropriate call as defined in the corresponding eLearning standard. This process tells the SAP SuccessFactors Learning application to end the content session, and SAP SuccessFactors Learning uses it as a trigger for storing the final time value received from the content.

5.9 SCORM 1.2 to SCORM 2004: Sequencing

Although the transition from SCORM 1.2 to 2004 changed the standard in specific ways, the major change was to add sequencing.

SCORM 2004 introduced some changes to the Run Time Environment (RTE). Most of it is terminology changes. The primary change with SCORM 2004 is the addition of a Sequencing and Navigation standard. This allows course developers to detail how a learner may proceed through the course structure. The sequencing rules for a SCORM 2004 course are contained within the course manifest.

Related Information

SCORM 2004 Sequencing [page 22]

5.9.1 SCORM 1.2 to 2004 Run Time Environment (RTE) Changes

SCORM 2004 RTE changed `cmi.core.lesson_status`, time formats, and error codes.

- In SCORM 1.2, the `cmi.core.lesson_status` data element was used to indicated SCO mastery and completion. SCORM 2004 replaced the `cmi.core.lesson_status` with two new data elements: `cmi.completion_status` and `cmi.success_status`. The `completion_status` element is now used to indicate SCO completion and the `success_status` element indicates SCO mastery.
- The `timespan` format changed from `HHHH:MM:SS.SS` to `P#Y#M#DT#H#M#S` so `P1Y3M2DT20H30M22.34S` indicates a period of 1 year, 3 months, 2 days, 20 hours, 30 minutes, and 22.34 seconds. The number of seconds is limited to two decimal places. Likewise the `timestamp` format changed from `HH:MM:SS.SS` to `YYYY-MM-DDTHH:MM:SS.SSTZD`, for example, `2006-04-07T13:15:55.32EST`.
- SCORM 2004 added a new error codes. Several of the SCORM 2004 error codes differ from the same codes in SCORM 1.2.

5.9.2 SCORM 1.2 to 2004 Run Time Environment (RTE) Function Name Changes

Between SCORM 1.2 and SCORM 2004, the RTE function names changes.

SCORM 1.2	SCORM 2004
LMSInitialize	Initialize
LMSGetValue	GetValue
LMSSetValue	SetValue
LMSCommit	Commit
LMSFinish	Finish
LMSGetLastError	GetLastError
LMSGetErrorString	GetErrorString
LMSGetDiagnostic	GetDiagnostic

5.9.3 SCORM 1.2 to 2004 Run Time Environment Additions

The following data elements were added to the SCORM Run Time Environment (RTE) from SCORM 1.2 to SCORM 2004.

New Data Element	Description
`cmi.completion_status`	(read, write) - Indicates whether the learner has completed the SCO - vocabulary: completed, incomplete, not_attempted, unknown.
`cmi.progress_measure`	(read, write) - Measure of the progress the learner has made toward completing the SCO - value from 0.0 to 1.0. Used in conjunction with `cmi.completion_threshold` for LMS determination of SCO completion.
`cmi.completion_threshold`	(read) - Progress measure at which the LMS should consider the SCO complete. - value from 0.0 to 1.0 - Used in conjunction with `cmi.progress_measure` for LMS determination of SCO completion.
`cmi.success_status`	(read, write) - Indicates whether the learner has mastered the SCO - vocabulary: passed, failed, unknown.

New Data Element	Description
`cmi.score.scaled`	(read, write) - Score value scaled to fit the range –1.0 to 1.0 inclusive. Used in conjunction with cmi.scaled_passing_score for LMS determination of SCO mastery.
`cmi.scaled_passing_score`	(read) - The scaled passing score required to master the SCO. The value of the data model element is scaled to fit the range -1 to 1 inclusive. Used in conjunction with `cmi.score.scaled` for LMS determination of SCO mastery.
`cmi.comments_from_learner.n.comment`	(read, write) - Used the capture a learner comment, where "n" is the comment number.
`cmi.comments_from_learner._children`	(read) - Returns the vocabulary for the cmi.comments_from_learner - returns: comment, location, timestamp.
`cmi.comments_from_learner._count`	(read) - Returns the number of comments currently stored for the LMS for this SCO.
`cmi.comments_from_learner.n.location`	(read, write) - Used to identify the location in the SCO to which the learner comment applies.
`cmi.comments_from_learner.n.timestamp`	(read, write) - Date and time the comment was created.
`cmi.comments_from_lms.n.comment`	(read, write) - Used to present a comment from the LMS.
`cmi.comments_from_lms._children`	(read) - Returns the vocabulary for the `cmi.comments_from_learner` - returns: comment, location, timestamp.
`cmi.comments_from_lms._count`	(read) - Returns the number of comments currently stored for the LMS for this SCO.
`cmi.comments_from_lms.n.location`	(read, write) - Used to identify the location in the SCO to which the LMS comment applies.
`cmi.comments_from_lms.n.timestamp`	(read, write) - Date and time the comment was created.
`cmi.objectives.n.score.scaled`	(read, write) - Objective score value scaled to fit the range –1.0 to 1.0 inclusive.
`cmi.objectives.n.success_status`	(read, write) - Indicates whether the learner has mastered the objective - vocabulary: passed, failed, unknown.
`cmi.objectives.n.completion_status`	(read, write) - Indicates whether the learner has completed the objective - vocabulary: completed, incomplete, not_attempted, unknown.
`cmi.objectives.n.progress_measure`	(read, write) - Measure of the progress the learner has made toward completing the objective - value from 0.0 to 1.0.

New Data Element	Description
`cmi.objectives.n.description`	(read, write) - Description of the objective.
`cmi.interactions.n.description`	(read, write) - Description of the interaction.
`adl.nav.request`	(read) - Used to indicate the navigation command the LMS should initiate when the SCO id terminated - vocabulary: continue, previous, choice, exit, exitAll, abandon, abandonAll
`adl.nav.request_valid.continue`	(read) - Used to determine if "continue" is a valid navigation request for the SCO.
`adl.nav.request_valid.previous`	(read) - Used to determine if "previous" is a valid navigation request for the SCO.
`adl.nav.request_valid.choice.{target=STRING}`	(read) - Used to determine if "choice" is a valid navigation request for the SCO, where `target=STRING` indicates the desired SCO.

5.9.4 SCORM 1.2 to 2004 Core Data Group Changes

Some of the data groups and individual data elements were renamed and some data groups were dropped. The core data group was dropped so the `cmi.core.xxx` data elements were changed to `cmi.xxx`. For example, `cmi.core.score.raw` changed to `cmi.score.raw`. The `cmi.core._children` data element was removed.

Changed Data Groups

SCORM 1.2	SCORM 2004
cmi.core.xxx	cmi.xxx
cmi.core.lesson_location	cmi.location
cmi.core.lesson_mode	cmi.mode
cmi.core.student_id	cmi.learner_id
cmi.core.student_name	cmi.learner_name
cmi.student_data.xxx	cmi.xxx
cmi.student_preference.xxx	cmi.learner_preference.xxx
cmi.student_preference.text	cmi.learner_preference.audio_captioning
cmi.student_preference.audio	cmi.learner_preference.audio_level
cmi.student_preference.speed	cmi.learner_preference.delivery_speed

SCORM 1.2	SCORM 2004
cmi.comments	cmi.comments_from_learner
cmi.interactions.n.student_response	cmi.interactions.n.learner_response
cmi.interactions.n.time	cmi.interactions.n.timestamp

5.9.5 SCORM 1.2 to 2004 Content Aggregation Model Changes

SCORM 2004 added new data elements to support its sequencing and navigation standard.

New Parent Elements for Sequencing

New Parent Element	Description
`Sequencing`	Defined for `item` or `organization` elements. The sequencing element details all of the sequencing information for a given activity.
`Presentation`	Defined for `item` elements that reference a SCO. The presentation element allows a course designer to remove LMS navigation controls for a particular SCO.

New Elements Added for Content Aggregation

New Element	Description
`completionThreshold`	Defined for `item` elements that reference a SCO. Used by the LMS to set the `cmi.completion_threshold` value for the SCO.
`minNormalizedMeasure`	Defined for `primaryObjective` element under the sequencing elements for a SCO. Used by the LMS to set the `cmi.scaled_passing_score` value for the SCO.

5.10 Characters Not Supported in Content File Names

If you have trouble launching your content, then there is a possibility that you have used some unsupported special characters in content file names.

The launch URL is created based on the imported content zip file. To avoid an error launching content, please ensure all files in the import zip file and any content package ID contain only supported characters. SAP SuccessFactors Learning follows the IETF standards while adding file names to the content to be imported. The links below lists down the supported characters in file names.

- https://www.ietf.org/rfc/rfc1738.txt
- https://tools.ietf.org/html/rfc3986

> **Note**
>
> Polish special characters, such as **ą** and **ć**, are **not** supported.

6 Cross Domain Solution Troubleshooting

This topic describes how you can troubleshoot issues with your Cross Domain Solution (CDS) configuration.

Turning On Debugging for All User Sessions

The easiest but broadest way to debug the CDS is to turn on debugging in LMS_ADMIN. When you turn on debugging in this way, all user sessions are put into debugging mode. We recommend this approach if you're debugging a staging server. Go to *System Admin* > *Configuration* > *System Configuration* > *LMS_ADMIN*. Find the `contentTestPagesEnabled` setting and change its value to true.

Checking the Tomcat Logs for Errors

First, check the Tomcat Access Log. This lets you know if the call came to the proxlet. Next, check the Tomcat StdErr Log and/or Tomcat localhost log. This shows you if the proxlet application generated any errors. The last step is to check the Catalina log. This lets you know if the Tomcat service started successfully.

Checking Proxlet Communication

To determine if the content is communicating correctly with proxlet, you can enable debug for the client session in a Chrome browser. After the user goes to the Learning Plan page, open a new tab and copy and paste this URL:`https://<customer_domain>/learning/tools/client-logging-config.jsp`, replacing `<customer domain>` with the appropriate value for your environment. On this page, you can enable debugging and then you can go back to the Learning Plan page and launch the course. Once the course is launched, open the developer's tools and then go the Javascript Console. The content communication debug messages should help in identifying the problem.

Capturing Fiddler Logs

The Fiddler logs can be used to review the response content for the `CrossDomainProxlet` calls. JCE encryption must be enabled.

6.1 Using Test Pages to Troubleshoot Communication

Use a test page to troubleshoot communication if you suspect that communication is failing.

Prerequisites

We recommend that you first run a simple test to see if the content server is communicating the most common data back to the LMS. Create a user, assign the user the troubled course from the content server, and see if the user gets data back from the server.

Procedure

1. Get the cross domain .zip file.
2. Make sure that the pages in `content.zip/test` are deployed.

   We also recommend that you look at the pages in `/test` so that you can see the available pages.
3. For the tracking that you want to troubleshoot, go to that page in `/test`.

   For example, go to `<your content server>/test/scorm2004_testpage.html`. Please note that the exact location of the test pages can differ based on how you deployed the files on your content server.
4. Use the buttons on the page to test the API calls to make sure that communication is functioning correctly.
5. Use the fields to pass data to the API calls.

Common problems you might uncover and solve:

- If the content isn't hosted on the LMS server, contact the vendor to make sure there are no restrictions as to where the content can be hosted.
- If the content is hosted outside of your firewall, ask the vendor if the content communication originates from the student's client machine or the content server. If the server is communicating for the content, the communication may be blocked by your firewall.

6.2 Cross Domain Solution at Run Time

Users are **not** aware of the Cross Domain Solution (CDS) when they interact with it at run time, but they are working with it to earn credit for their progress. This topic describes how they are interacting with the CDS.

When users click a link in their learning assignments to launch content, they open a new window that is served from the content server. The users see their content in a separate window and do not know about the wrapper.

- Inside the wrapper window, users navigate the course, exam, or other content. This content is served from the content server.

- While navigating the content, the user's progress is tracked with JSlet (JavaScript "applet") or a Java Applet tracking code.
- The proxlet.war application relays the user's progress to the LMS.

7 Configuring Cross Domain Solutions

7.1 Getting the Latest Proxlet WAR for Cross-Domain Content (CDC)

Get the latest Proxlet WAR and other CDC files so that you can keep your third party content server in sync with SAP SuccessFactors Learning.

Prerequisites

Before you download the latest files, ensure that your content server meets the current technical requirements provided in Technical Requirements for the Cross Domain Solution (CDS) [page 95].

Context

The Cross-Domain Content (CDC) solution is for customers who implement their own content server instead of using premium content management. We sometimes change SAP SuccessFactors Learning in a way that requires a change to content servers. If you are a premium content management customer, you get the change automatically. If you run your own third party solution, however, you must keep your third party content server in sync with the SAP SuccessFactors Learning server so that calls from the client are recognized by the server.

Procedure

1. Go to Support Downloads and navigate to the *Installations and Upgrades* page.
2. Choose *Installation*.
3. Download the cross domain proxlet ZIP file.

Next Steps

The ZIP file contains many files that you need for your third party content server, including the Proxet WAR file, the content.zip file, which contains the content launch wrapper files, and troubleshooting pages. You need to deploy the WAR file and complete any release-specific steps to maintain your content-hosting server.

7.2 Setting Up the Cross Domain Solution on a Content Server

If you choose to host your own Learning content, you set up the Cross Domain Solution (CDS) on your content-hosting server so that the Learning application can communicate with your hosting server. This enables learners to take courses, track progress, and receive completion credit for self-hosted content.

Prerequisites

- Identify or create a web accessible directory in your content-hosting server to contain the content launch wrapper files. We recommend `/content`.
- Download the `crossdomain.zip` file.
- Identify your company's Learning URL.
- Identify the URL for the common super domain.
- Review the Technical Requirements for the Cross Domain Solution (CDS) [page 95] information and ensure your content-hosting server meets the requirements.

Procedure

1. Extract the files from `crossdomain.zip` to a temporary folder.
   You see two files in the temporary folder: `content.zip` and `proxlet.war`.
2. Extract the files from `content.zip` to the web accessible directory.

   We recommend `/content`.
3. In your Java Virtual Machine (JVM), set the argument `-DLMSServerWhitelist` to a comma-separated list of the Learning servers with which your content-hosting server needs to communicate, including the common super domain. For example, `-DLMSServerWhitelist=https://host.domain.com,https://stage-host.domain.com,https://host.lms.hr.cloud.sap,https://stage-host.lms.hr.cloud.sap`. This is required to launch self-hosted content from Learning.

   The argument can be set in one of two ways, depending on your configuration:

   - Edit the `<TOMCAT_HOME>/bin/setenv.sh` file and set the `CATALINA_OPTS` value with the argument.
   - Edit the `<TOMCAT_HOME>/bin/Catalina.sh` file and update with the argument.

   > **Note**
   >
   > It's recommended that you have one proxlet servicing a single Learning server. If you need to use one proxlet to communicate with multiple Learning servers, you'll need to list all the Learning servers here.
4. In the same location as step 3, set the argument `-Dhttps.protocols` to a comma-separated list of the TLS protocols.

   ```
   -Dhttps.protocols=TLSv1.2,TLSv1.3
   ```

5. Deploy `proxlet.war` on the content-hosting server, according to the instructions for Tomcat server or servlet container.

Technical Requirements for the Cross Domain Solution (CDS) [page 95]
If you install the Cross Domain Solution so you can host your own Learning content, your content-hosting server must meet these standards.

Proxlet Cross Domain Solution Overview [page 95]
This topic describes how the Cross Domain Solution (CDS) proxlet helps the system track users' course completion when the content server is on a different domain from SAP SuccessFactors Learning.

7.2.1 Technical Requirements for the Cross Domain Solution (CDS)

If you install the Cross Domain Solution so you can host your own Learning content, your content-hosting server must meet these standards.

CDS Requirements

Category	Requirements
Web Server	• Latest Tomcat 9 version
Java	• Latest JVM 17 version (openjdk version 17.0.9) • JCE unlimited cryptography enabled **Note** To verify, go to `$JAVA_HOME/jre/lib/security` and edit the java.security file. The `crypto.policy=unlimited` line needs to be in the file and it can't be commented.

Parent topic: Setting Up the Cross Domain Solution on a Content Server [page 94]

Related Information

Proxlet Cross Domain Solution Overview [page 95]

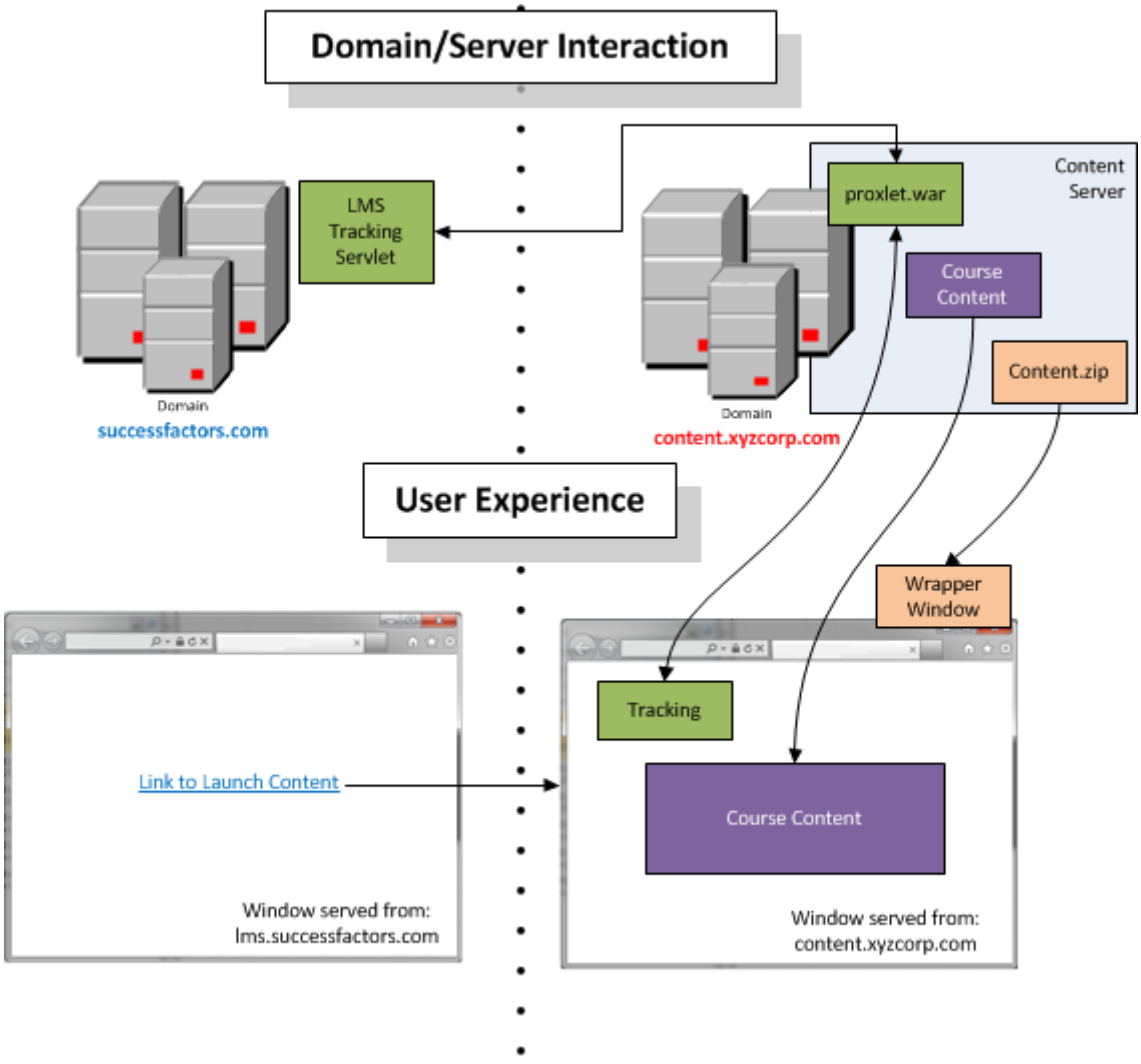
7.2.2 Proxlet Cross Domain Solution Overview

This topic describes how the Cross Domain Solution (CDS) proxlet helps the system track users' course completion when the content server is on a different domain from SAP SuccessFactors Learning.

The SAP SuccessFactors Learning Cross Domain Solution (CDS) provides trusted server-to-server communication between an outside content server and SAP SuccessFactors Learning. The trusted communication is necessary

when the course content server is in a different Domain Name Space (DNS) than the SAP SuccessFactors Learning DNS. This is common.

In a cross domain scenario, SuccessFactors recommends the Proxlet WAR CDS. The Proxlet WAR (Java Web ARchive) communicates basic course tracking information back to SAP SuccessFactors Learning, such as exam scores and progress through content. When the Proxlet WAR CDS is NOT working properly, the user can launch content, but their completion status is not recorded back to SAP SuccessFactors Learning.



Proxlet Cross Domain Solution

> **Note**
>
> SAP SuccessFactors Learning has a second CDS called a **Cookie CDS**. The cookie CDS is less common, so this document refers to the Proxlet WAR CDS as "the CDS."

Parent topic: Setting Up the Cross Domain Solution on a Content Server [page 94]

Related Information

Technical Requirements for the Cross Domain Solution (CDS) [page 95]

7.3 Configuring LMS_ADMIN for the Cross Domain Solution

You configure the LMS_ADMIN file to point the Learning Management System (LMS) to the content server where you deployed the Cross Domain Solution (CDS).

Prerequisites

You have set up the CDS on the content server. See Setting Up the Cross Domain Solution on a Content Server [page 94]

Procedure

1. Log in to Learning Administration as a system administrator.
2. Go to *System Administration* > *Configuration* > *System Configuration* > *LMS_ADMIN* .
3. Edit the communication CDS settings. See Communication Cross Domain Support Settings [page 98].

Related Information

Communication Cross Domain Support Settings [page 98]
Cross Domain Solution Troubleshooting [page 90]

7.3.1 Communication Cross Domain Support Settings

Use these settings in LMS_ADMIN to configure Cross Domain Support (CDS) to the proxlet.war.

Properties

> **Note**
>
> Make sure that you do not accidentally enable cookie CDS. The cookie cross domain solution is not as common.

Property	Recommended Value	Description
communicationCrossDomainSupportEnabled	true	Set to true to turn on the CDS: it enables communication with the proxlet.
defaultContentServerPath	<content server URL root>/content	The content server wrapper page path is the directory on your content server where you extracted the wrapper files. It must be an absolute URL.
contentWrapperPageName	main_content_wrapper.html	This is the main content wrapper page that you extracted. Do not change.
scorm2004ContentWrapperPageName	scorm_2004_content_wrapper.html	This is the main content wrapper page that you extracted for SCORM. Do not change.
externalURLQuestionWrapperPageName	main_question_wrapper.html	This is the main content wrapper page that you extracted for questions. Do not change.
contentOpenerPageName	main_content_opener.html	This is the main content opener page that you extracted. Do not change.
defaultProxletPath	<app server root URL>/proxlet	This is the path to the proxlet WAR that you deployed. Must be an absolute URL.
lmsSessionId	JSESSIONID	Do not change.

Configuration Example

```
communicationCrossDomainSupportEnabled=true
defaultContentServerPath=http://content.xyzcorp.com/content
contentWrapperPageName=main_content_wrapper.html
```

```
scorm2004ContentWrapperPageName=scorm_2004_content_wrapper.html
externalURLQuestionWrapperPageName=main_question_wrapper.html
contentOpenerPageName=main_content_opener.html
defaultProxletPath=http://content.xyzcorp.com:8080/proxlet
lmsSessionId=JSESSIONID
```

Related Information

Configuring LMS_ADMIN for the Cross Domain Solution [page 97]
Enabling the Cross Domain Solution on Content Objects [page 99]
Cross Domain Solution Troubleshooting [page 90]

7.4 Enabling the Cross Domain Solution on Content Objects

You enable individual content objects for the Cross Domain Solution (CDS) so that the content object can work properly in the CDS context.

Prerequisites

- You configured LMS_ADMIN and know the values you set for the properties.
- You have imported or created the content objects you want to configure.

Procedure

1. Go to Go to Learning Administration and then go to *Content* > *Content Objects*.
2. Find your content object, and then open *Cross Security Domain*.
3. Select *Enable Communication Cross Domain through Proxy* and then click *Apply Changes*.
4. Check the settings in *Cross Domain Settings*.

   You see default values based on your Cross Domain Solution (CDS) configuration. In most cases, you do not need to make changes.

   Clear *Use Frameset* if your content doesn't run well when launched under a frameset.

Related Information

Configuring LMS_ADMIN for the Cross Domain Solution [page 97]

Cross Domain Solution Troubleshooting [page 90]

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

