



Advanced Distributed Learning Initiative

Sharable Content Object
Reference Model (SCORM™)

Version 1.2

SCORM ランタイム環境

2001 年 10 月 1 日

このページは空白である.

Advanced Distributed Learning
Sharable Content Object Reference Model
バージョン 1.2
SCORM ランタイム環境

ADLNet から入手可能
(<http://www.adlnet.org/>)

日本語版は ALIC, eLC から入手可能
(<http://www.alic.gr.jp/>, <http://www.elc.or.jp/>)

質問やコメントは ADLNet 問い合わせセンタまで。
日本語版に関する質問やコメントは ALIC, eLC まで。

このページは空白である.

エディタ
Philip Dodds (ADL)

主要貢献エディタ (ADL)

Ron Ball
William Capone
Jeff Falls
Dexter Fletcher
Alan Hoberney
Paul Jesukiewicz
Kirk Johnson
Mary Krauland

Jeff Krinock
Lori Morealli
Douglas Peterson
Jonathan Poltrack
Chris Snyder
Schawn Thropp
Bryce Walat
Jerry West

貢献者リストの一部:

**Alliance of Remote Instructional Authoring & Distribution
Networks for Europe (ARIADNE) (<http://www.ariadne-eu.org/>)**

Erik Duval
Eddy Forte
Florence Haenny
Ken Warkentyne

**Aviation Industry CBT (Computer-Based Training) Committee (AICC)
(<http://www.aicc.org/>)**

Jack Hyde
Bill McDonald
Anne Montgomery

**Institute of Electrical and Electronics Engineers (IEEE)
Learning Technology Standards Committee (LTSC) (<http://ltsc.ieee.org/>)**

Mike Fore
Wayne Hodgins

IMS Global Learning Consortium, Inc. (<http://www.imsglobal.org/>)

Thor Anderson
Steve Griffin
Mark Norton
Ed Walker

(At Large)

Bob Alcorn
Lenny Greenberg
Chris Moffatt
Boyd Nielsen
Claude Ostin
Chantal Paquin

Mike Pettit
Dan Rehak
Tom Rhodes
Tyde Richards
Roger St. Pierre
Kenny Young

...and many others.

日本語版翻訳者

ALIC 相互運用性部会コンテンツ標準 WG ガイドライン SWG eLC 相互運用性委員会ガイドライン WG

氏名	所属	担当
石打 智美	NTT データ (株)	2 章, 査読
関口 法門	(株) 富士ゼロックス総合教育研究所	査読
高橋 和彦	(株) 富士通インフォソフトテクノロジー	3 章, 査読
田口 正弘	(株) アクセスチケットシステムズ	査読
樋田 稔	エスエイティーティー (株)	1 章, 査読
仲林 清	(株) エヌ・ティ・ティ エックス	1 章, 2 章, 査読
廣瀬 文男	ウイルソンラーニングワールドワイド (株)	査読

このページは空白である.

目次

3 章 SCORM™ ランタイム環境.....	3-1
3.1. ランタイム環境の概要	3-3
3.2. 起動	3-5
3.2.1. 学習資源の起動	3-5
3.2.1.1. アセット	3-6
3.2.1.2. Sharable Content Object(SCO).....	3-6
3.3. アプリケーションプログラムインターフェース (API).....	3-7
3.3.1. API の概要	3-7
3.3.2. SCO/LMS 間通信の記述.....	3-7
3.3.2.1. SCO/LMS 間通信 API の詳細.....	3-8
3.3.2.2. SCO / LMS 間通信の API アダプタ状態遷移	3-13
3.3.3. API エラー・コードの使用法	3-15
3.3.4. API の一般規則	3-17
3.3.5. LMS の責任範囲	3-17
3.3.5.1. API アダプタ	3-17
3.3.6. SCO の責任範囲.....	3-18
3.3.6.1. Find API	3-18
3.4. データモデル	3-20
3.4.1. データモデルの概要	3-20
3.4.1.1. SCORM ランタイム環境データモデル	3-20
3.4.1.2. SCORM ランタイム環境データモデルの一般規則	3-21
3.4.2. データモデル要素	3-21
3.4.3. リストの扱い	3-22
3.4.4. SCORM ランタイム環境データモデル	3-23
3.4.5. データタイプおよび制御ボキャブラリ	3-85
3.5. ランタイム環境の動作	3-89
APPENDIX A 略語リスト	A-1
APPENDIX B 参考文献	B-1
APPENDIX C 改訂履歴	C-1

このページは空白である.

3 章

SCORM™ ランタイム環境

このページは空白である.

3.1. ランタイム環境の概要

SCORM の目的は、学習資源を再利用可能とし、複数の学習管理システム(LMS)にわたって相互運用可能とすることである。これを可能とするために、学習資源を起動する共通の方法、学習資源が LMS と通信するための共通のメカニズム、そして、通信の基礎を形成するための事前に定義された言語、ないし、ボキャブラリがなければならない。図 3.1a に図示するように、ランタイム環境の 3 つの側面は起動、応用プログラムインターフェース（以下、API）およびデータモデルである。

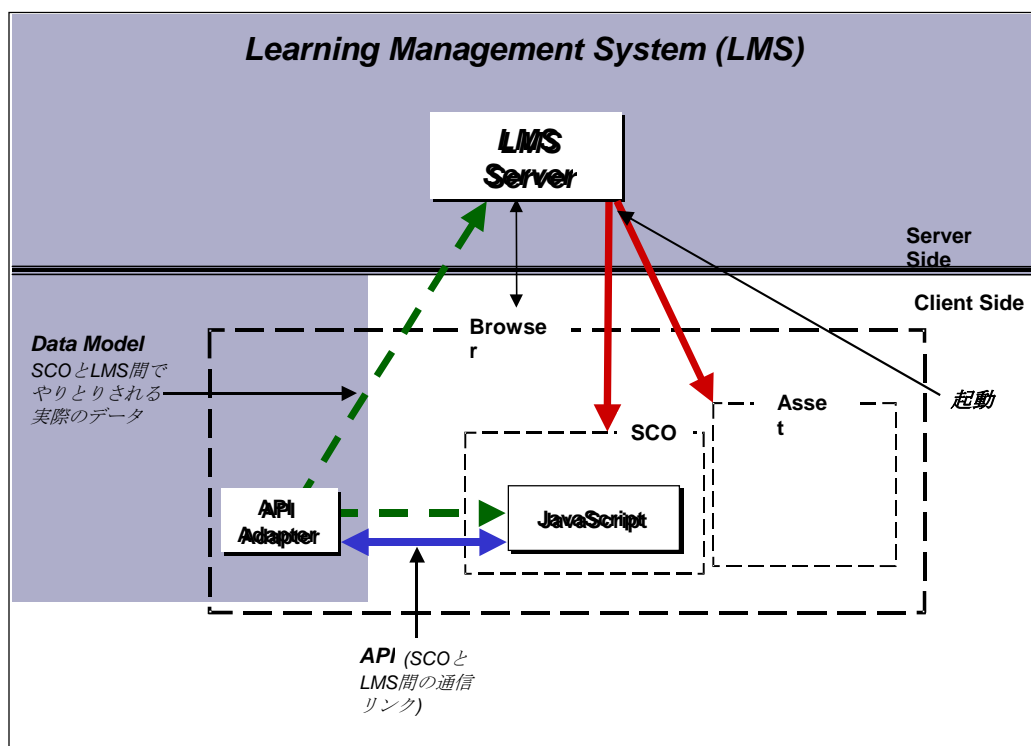


図3.1a:SCORM ランタイム環境に適用する起動, API およびデータモデル

起動メカニズムは、LMS が Web ベースの学習資源を起動するための共通な方法を定義する。このメカニズムは、配信された学習資源と LMS の間で通信を確立するための手続きと責任範囲を定義する。通信プロトコルは、共通の API の使用を通して標準化される。

API は、LMS に学習資源の状態(例えば、初期化、終了、エラーの条件)を知らせる通信メカニズムであり、LMS と Sharable Content Object(SCO)の間でデータ(例えば得点、制限時間、など)を取得したり、設定したりするために使用される。

データモデル(Data Model)は、伝達される(例えば学習内容の状態)情報を定義するために使われるデータ要素の標準セットである。最も単純な場合、データモデルは、LMS と SCO の両方が「知っている」ことになっている要素を定義している。LMS は要求されるデータ要素の状態を複数セッションに渡って維持しなければならない。さらに、学習コンテンツは、複数のシステム間にまたがる再利用が発生する場合、これらのあらかじめ定義されたデータ要素だけを利用しなければならない。

3.2. 起動

起動の共通機構は，その基盤である LMS の実装の依らずに，複数の LMS 間で学習資源の配信動作の整合を可能にする．この文脈では，“LMS”という用語は，学習資源の配信管理機能を含むシステムを記述するために使用していることに注意されたい．この起動の機構は，学習経験の文脈で，コンテンツオブジェクト(SCOs)とアセットの形式でウェブ対応の学習資源の配信を扱うものである．

3.2.1. 学習資源の起動

コンテンツアグリゲーションモデルで示されたように，SCORM コンテンツモデルは，次の3つの構成要素で構成される．

- アセット
- SCO
- コンテンツアグリゲーション

LMS によって起動され得る 2 つの SCORM コンテンツモデル構成要素は，アセット，及び SCO である．起動される学習資源のタイプに応じて異なる起動要求がある．起動メカニズムは，LMS が学習資源を起動する共通の方法を定義する．配信された学習資源と LMS 間の通信を確立するための手続きと責任範囲は，起動された SCORM 学習資源のタイプに応じて異なる．

コンテンツパッケージで定義されたコンテンツ構造に基づく学習資源の間のシーケンシング，とナビゲーションを管理することは，LMS の役割である．LMS は，学習資源に定義された前提条件の達成に基づいて適応的にシーケンシングを決定する．個別の学習経験を構成する学習資源を通しての進行は，LMS の能力に依存していて，シーケンシャル，非シーケンシャル，ユーザ指示，あるいは，学習者適応のいずれかとなる．現時点で，SCORM は，学習資源の間のシーケンシングとナビゲーションの標準化を扱わない．これはまた，シーケンシングに関係したビジュアルコンポーネントのルック・アンド・フィールのガイドラインが存在しないことを意味する．これは，SCORM の将来のバージョンで処理されるであろう．

例えば，LMS は，コンテンツアグリゲーション内をユーザ主導でナビゲーションするためのメニューを表示することができる．そのメニューは，一連のハイパーリンクで，ハイパーリンクのターゲットはメニューに現れる学習資源の対応する起動場所を含む．

他の方法として、LMS は、学習者のパフォーマンスに基づいて学習資源を起動する順番を適用的に決定するサーバサイドコンテンツ配信機能を含むか、利用することができる。この場合、配信機能を実行するサーバサイド構成要素は、直接、もしくは、間接的に適切な学習資源の指定された起動場所に基づいて、その学習資源を提供するであろう。

なんらかのイベントに基づいて、どの学習資源を起動するかを決定するのは、LMS (あるいは配信機能/サービス)の役割である。LMS は、コンテンツ構造において定義された順に次の学習資源を起動するか、ユーザが選択した SCO を起動するか、あるいは、適応的な方式により学習者パフォーマンスに基づき、起動する SCO を決定する。起動する適切な学習資源を決定する際、LMS はコンテンツパッケージ中で定義された学習資源の起動場所によって定められる URL を使って、その場所に存在する学習資源に誘導するか、あるいは、その場所に存在する学習資源と現在表示された学習資源を置換する。

LMS は、起動機能を望みの任意の方法で起動を実装してもよいし、必要に応じて LMS のクライアントまたはサーバ部分に実際の起動機能を任せてもよい。現実の起動は、HTTP プロトコルを用いて果たさなければならない。最終的に、内容パッケージの起動ロケーションによって指定された学習資源は、起動されクライアントブラウザに渡される。

3.2.1.1. アセット

アセットに対応する学習資源の場合、SCORM 起動モデルが要求するのは、LMS が HTTP プロトコルを使ってアセットを起動することだけである。アセットは、API とデータモデルを使って LMS に通信する必要がないので、LMS によって供給された API アダプタを検索する必要性はない。

3.2.1.2. Sharable Content Object(SCO)

SCO に対応する学習資源の場合、SCORM 起動モデルが要求するのは、LMS が一度に一つの SCO のみを起動し、一つの SCO だけが同度にアクティブになることである。SCO は他の SCO を起動してはならない。

LMS は、SCO を、LMS ウィンドウの子ウィンドウまたは子フレームであるブラウザ・ウィンドウで起動しなくてはならない。ここで、LMS ウィンドウは API アダプタをドキュメント・オブジェクト・モデル(DOM)²⁷ オブジェクトとして公開する。API アダプタは LMS により提供されなければならない。

SCO は API アダプタが見つかるまで、親およびオープンウィンドウの階層を再帰的に探索しなくてはならない。ひとたび API アダプタが見つければ、SCO は LMS とコミュニケーションを開始できる。

3.3. アプリケーションプログラムインターフェース (API)

3.3.1. API の概要

SCORM は、AICC の CMI001Guidelines for Interoperability⁴ ドキュメントの中で定義された、ランタイム環境機能に直接基づいている。ADL は、AICC メンバおよび参加者と協力して共通の起動、API 仕様を開発し、ウェブ・ベースのデータ要素を採用した。次のセクションでは、SCORM に関係するものとして、AICC API 仕様の重要な要素の概観を提示する。

3.3.2. SCO / LMS 間通信の記述

共通 API を使用することにより、相互運用性と再利用に関する多くの SCORM 高水準要求を満たすことができる。共通 API は、SCO が LMS と通信するための標準化された方法を提供し、同時に、コンテンツ開発者に対して通信機能の個別の実装を隠蔽する。最も単純な言い方をすれば、API は単なる既定の機能セットであり、SCO は利用可能なものとしてそれに依存することができる。API は SCO から実装の詳細を隠蔽し、それにより再利用と相互運用性を促進する。API アダプタは API の機能を実現・提示する機能ソフトウェア部品である。コンテンツ開発者が同じ公開インターフェースを使用する限り、API アダプタの内部実現方法はコンテンツ開発者が意識すべき事項であってはならない。LMS は単に、API の機能を実現し、クライアント SCO にそのインターフェースを提示する API アダプタを提供する必要がある。

API の重要な面は、SCO が LMS と通信することを可能にするコミュニケーション・メカニズムであるということである。ひとたび SCO が起動されたならば、SCO は LMS と情報を “get” したり “set” したりする。API アダプタと SCO 間の全てのコミュニケーションは SCO によって開始される。現在、LMS から、SCO が実現する機能と呼び出し開始するためのサポート機能はない。API アダプタ (API Adaptor) オブジェクトの機能は以下の 3 つである。

- 実行状態

API 機能 LMSInitialize(“ ”) および LMSFinish (“ ”) の 2 つは、実行状態を扱う。

- 状態管理

API は、エラーを扱うために使用される 3 つの機能を持つ。これらの 3 つの API 機能は次のとおりである: LMSGetLastError(), LMSGetErrorString(errornumber) および LMSGetDiagnostic(parameter)。

● データ転送

他の 3 つの API 機能は LMS との間でのデータのやりとりに使用される: `LMSGetValue(data model element)`, `LMSSetValue(data model element, value)` および `LMSCommit(“”)`. ここで, API は, 外部データモデルとして別個に定義されたデータ値の取得, 設定を行うよう設計されていることに注意されたい. AICC 仕様は, このようなデータ・モデルのひとつで “cmi” と呼ばれるものを定義している. 他のデータ・モデルも, 同様に開発され, この API とともに使用することができる.

3.3.2.1. SCO / LMS 間通信 API の詳細

次のテーブルは SCO から LMS への交信 API を詳細に定義する.

状態実行	
LMSInitialize	<p>記述: この機能は, API アダプタに対して, SCO がこれから LMS と通信することを示す. これにより, LMS は, LMS 固有の初期化項目を取り扱うことができる. SCO は, 他のどの API 機能と呼ぶよりも前にこの機能と呼ばなくてはならない.</p> <p>構文: <code>LMSInitialize(パラメータ)</code></p> <p>パラメータ: “” この標準に適合するためには, 空文字列(“”)を渡されなければならない. 空文字列(“”)以外の値は将来の拡張のために予約されている.</p> <p>返り値: 文字列は真理値を表わす.</p> <ul style="list-style-type: none">.. 「True」結果は, <code>LMSInitialize(“”)</code> が成功したことを示す... 「False」結果は, <code>LMSInitialize(“”)</code> が失敗だったことを示す. <p>「False」の返り値が返された場合, LMS が未知の状態であり, 以降のいかなる API 呼び出しも LMS によって処理されないことを SCO に示す.</p> <p>例:</p> <pre>var result = LMSInitialize(“”) if (result == "false") { // Do some error handling } else { // continue with the execution of the SCO }</pre> <p>SCO は, コンテンツが LMS とのコミュニケーションを確立したいと API アダプタに伝える. 典型的な返り値は「True」.</p>
LMSFinish	<p>記述: SCO は, この時点より前に <code>LMSInitialize</code> の呼び出しに成功した後, LMS との交信が不要と判断すれば, この関数を呼びださなければならない.</p>

	<p>らない。この呼び出しは 2 つのことを意味する:</p> <ol style="list-style-type: none"> LMSSetValue()呼び出しを使用したいかなる設定データも LMS によって保持されることが SCO に対して保証される。 SCO は LMS との通信を終了した。 <p>構文: LMSFinish(parameter)</p> <p>パラメータ: "" この標準に適合するためには、空文字列("")を渡されなければならない。空文字列("")以外の値は将来の拡張のために予約されている。</p> <p>返り値: 真理値を表わす文字列</p> <ul style="list-style-type: none"> ・「True」結果は、LMSFinish("")が成功したことを示す。 ・「False」結果は、LMSFinish("")が失敗だったことを示す。 <p>「True」の返り値が返された場合、SCO は他の API 関数をそれ以上呼んではならない。</p> <p>「False」の返り値が返された場合、これは LMS が未知の状態であり、以降のいかなる API 呼び出しも、LMS によって処理されるか処理されないか不明であることを SCO に示す。</p> <p>例:</p> <pre>var result = LMSFinish("");</pre>
データの移行	
LMSGetValue	<p>記述: この機能により、SCO は LMS から情報を取得できる。この機能は以下を決定するために使用される:</p> <ul style="list-style-type: none"> ・データモデルの様々なカテゴリー(グループ)および要素の値 ・サポートされているデータモデルのバージョン ・特定のカテゴリーや要素がサポートされているかどうか。 ・配列や要素リスト中の現在のアイテム数 <p>完全なデータ要素名かキーワードがパラメータとして提供される。要求されたデータモデルパラメーターの現在値が返される。1 つの値だけ(常に文字列)が呼び出しごとに返される。</p> <p>構文: LMSGetValue(パラメータ)</p> <p>パラメータ:</p> <p>datamodel.group.element 指定された要素の値を返す。</p> <p>datamodel._version _version キーワードは LMS によってサポートされるデータモデルのバージョンを決定するために使用される。</p> <p>datamodel.element._count _count キーワードは現配列中の要素の数を決定するために使用される。数は配列中の最終位置の表示ではなく配列中の要素の総数である。</p> <p>datamodel.element._children _children キーワードは LMS によってサポートされるグループもしくはカテゴリ中すべての要素を決定するために使用される。</p>

	<p>返り値: 返り値はすべて文字列である.</p> <p>LMSGetValue(datamodel.group.element) 返り値は要求された要素あるいはグループの現在値を表わす列である.</p> <p>LMSGetValue(datamodel._version) 返り値は LMS によってサポートされたデータモデルのバージョンを表わす列である.</p> <p>LMSGetValue(datamodel.group._children) 戻り値は, LMS によってサポートされる, 指定のグループまたはカテゴリ中のすべての要素リスト名をコンマで区切ったものである. 要素に子はないが, 要素がサポートされる場合, 空文字列("")が返される. 要素がサポートされない場合も, 空文字列("")が返される. 要素がサポートされない場合, 引き続いて最後のエラーを要求する [LMSGetLastError()] ことで, 要求がサポートされているかどうかを決定できる. エラー「401 Not implemented error」は要素がサポートされないことを示す.</p> <p>LMSGetValue(datamodel.group._count) 返り値は現在の要素リストあるいは配列中のアイテム数を示す整数である.</p> <p>例: var value = LMSGetValue("cmi.core.student_name") 典型的な返り値は “Hyde,Jackson” といったものである. var value = LMSGetValue("cmi.core.lesson_status") 典型的な返り値は “incomplete”(非完了)である. var value = LMSGetValue("mi._version") CMI データモデルを定義する IEEE のドキュメントの現在の規格案は, Computer Managed Instruction 規格案というタイトルで, ID が P1484.11.2, バージョンが 3.4 である. この呼び出しは, IEEE のドキュメントのバージョン番号 3.4 を返す.</p> <p>var value = LMSGetValue("cmi.student_preferences._children")</p> <p>これはカテゴリサポート情報の要求である. 典型的な 1 つの返り値は “audio,speed,text” である. 空文字列(“ ”)が返される場合, student_preferences はサポートされていない. 最後のエラーを決定する追加の API 呼び出しでこれを確認できる.</p>
LMSSetValue	<p>記述: この機能により, SCO は LMS に情報を送ることができる. API アダプタは, LMS へ直ちに情報を転送するか, あるいは他のアプローチに基づいて情報を転送するように設計される. この機能はデータ・モデル中の様々なカテゴリー(グループ)や要素に現在値を設定するために使用される. データ要素名およびそのグループがパラメータとして提供される. 呼び出しの第 2 のパラメータとして, データ要素の新しく設定したい値が含まれる. 各呼び出しでは 1 つの値だけが送られる.</p> <p>構文: LMSSetValue(パラメータ,値)</p>

	<p>パラメータ: これはデータモデルに定義された完全適格要素名である。引数は大文字と小文字が区別される。引数は引用符で囲まれた文字列である。</p> <p>下記はこのパラメータがとり得るいくつかの形式を表わす。</p> <p>datamodel.element これはデータモデル(Data Model)の中で定義されたカテゴリあるいはグループの名前である。一つの例は“cmi.comments”である。</p> <p>datamodel.group.element これはデータモデル(Data Model)の中で定義された要素の名前である。一つの例は“cmi.core.lesson_status”である。</p> <p>datamodel.group.n.element 要素配列(ゼロベース方式のインデックス付けが使用される)の n - 1 番目のメンバのサブ要素の値。</p> <p>値: 第 1 パラメータで指定されるデータモデル要素に対して、この標準規格で定義されたデータ型に変換可能でなくてはならない文字列である。</p> <p>返り値: 文字列は真理値を表わす。 .. “True” 結果は、LMSSetValue()が成功したことを示す。 .. “False” 結果は、LMSSetValue()が失敗だったことを示す。</p> <p>例: var result = LMSSetValue("cmi.core.score.raw","95");</p> <p>cmi.core.score.raw に値 95 を設定 LMSGetValue("mi.core.score.raw")に続く呼び出しは、95 を返さなければならない。</p>
<p>LMSCcommit</p>	<p>記述: SCO から LMSSetValue()で受け取った値を API アダプタがキャッシングしている場合、この呼び出しは、LMS がまだ保持していないこれらすべての値を保持するように要求する。</p> <p>いくつかの実装では、API アダプタは設定値を受け取ると直ちに保持し、それらをクライアントにキャッシュしない。そのような実装では、この API 呼び出しは冗長で、API アダプタからは何の追加アクションも起きない。この呼び出しは、LMSSetValue()呼び出しによる送信データが LMSCcommit()の完了時に LMS によって保持されることを SCO に保証する。</p> <p>構文: LMSCcommit(パラメータ)</p> <p>パラメータ: "" この標準に適合するためには、空文字列("")を渡さなければならない。空文字列("")以外の値は将来の拡張のために予約されている。</p> <p>返り値: 真理値を表わす文字列 .. “True” 結果は、LMSCcommit("")が成功したことを示す。 .. “False” 結果は、LMSCcommit("")が失敗だったことを示す。 「False」の返り値が返された場合、これは LMS が未知の状態であり、以降のいかなる API 呼び出しも、LMS によって処理されるか</p>

	<p>処理されないか不明であることを SCO に示す.</p> <p>例: <code>var result = LMSCommit("");</code></p> <p>キャッシュされた任意の値は, SCO の LMSSetValue()呼び出しを経由して以前に設定されたもので, LMS によって持続されていない値が持続されるよう要求する.</p>
状態管理	
LMSGetLastError	<p>記述: SCO は, すべての API 呼び出しが成功したか否か, またもし不成功の場合, 何が悪かったかを評価する方法を持たねばならない. この関数は, 前の API 呼び出しに起因するエラー・ステータス・コードを返す. (この関数, LMSGetErrorString および LMSGetDiagnostic – すなわちエラー関数 – を除いて)API 関数が呼ばれるごとにエラーコードがリセットされる. SCO はエラーコードを検索するためにエラー関数を何回も呼び出すことができ, 次の API 呼び出しが行われるまで, そのコードは変わらない.</p> <p>構文: LMSGetLastError()</p> <p>パラメータ: 無し</p> <p>返り値: 返り値は文字列で, 以下のカテゴリに属するエラーを識別する整数番号に変換可能なものである.</p> <ul style="list-style-type: none"> 100 番台 一般的エラー 200 番台 構文エラー 300 番台 LMS エラー 400 番台 データモデルエラー <p>次のコードがエラー・メッセージに利用可能である.</p> <ul style="list-style-type: none"> 0 エラーなし 101 一般的な例外 201 不正引数エラー 202 要素は子要素を持たない 203 要素は配列でない – count を持たない 301 未初期化 401 非実装エラー 402 無効な値設定呼出し, 要素はキーワード 403 要素は読み出し専用 404 要素は書き込み専用 405 不正データ・タイプ <p>追加コードは未定義</p> <p>例: <code>var errorCode = LMSGetLastError();</code></p>
LMSGetErrorString	<p>記述: この関数により, コンテンツは, エラーコード番号によって表わされるエラーの記述を得ることができる.</p>

	<p>構文: <code>LMSGetErrorString(errornumber)</code></p> <p>パラメータ: エラーコードを表わす整数番号.</p> <p>返り値: エラー記述を表現する文字列.</p> <p>例: <code>var errorString = LMSGetErrorString("403");</code></p> <p>errorString には "Element is read only" (要素は読み出し専用) という文字列設定されなければならない.</p>
LMSGetDiagnostic	<p>記述: この機能はベンダー特有のエラー記述の開発及びコンテンツによるアクセスを可能にする. これらは, 通常, エラーに関する補足詳細を提供する.</p> <p>構文: <code>LMSGetDiagnostic(parameter)</code></p> <p>パラメータ: パラメータは, 2つの形式のうちの1つをとる. ..エラーコードを表わす整数番号. これは, リストされたエラーコードについての補足情報を要求するものである. ..""空文字列. これは, 発生した最後のエラーについての補足情報を要求するものである.</p> <p>返り値: 返り値は, 要求されたエラーあるいは最後のエラーのいずれかに関係する任意のベンダーに要望する追加の情報を表わす文字列である.</p> <p>例: <code>var moreInfo = LMSGetDiagnostic("403");</code></p> <p>moreinfo は, "Element is read only" (要素は読み出し専用) というエラーに関する, ベンダー特有の追加情報を含んでよい.</p>

3.3.2.2. SCO / LMS 間通信の API アダプタ状態遷移

SCO/LMS 間の通信 API アダプタは, 実行時に, SCO のインスタンス毎に, いくつかの状態(図 3.3.2.2a)を遷移する. API アダプタの状態により, 特定の入力イベントに対する API アダプタの応答が決まる. これらの各状態間において, SCO は異なるアクティビティを受けとる. API の遷移する状態は, "Not Initialize", "Initialized", "Finished" の3つである.

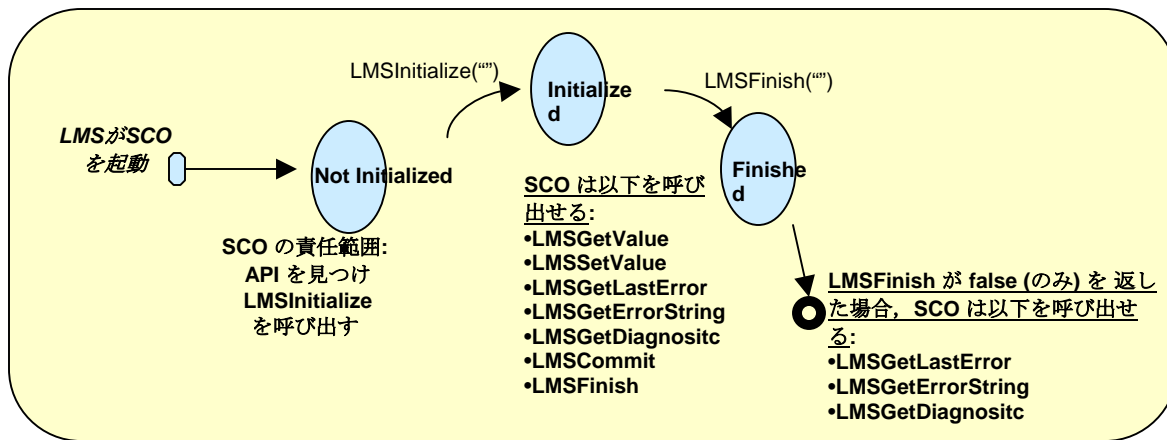


Figure 3.3.2.2a: API アダプタの状態遷移

Not Initialize(未初期化): これは、SCO が実際に起動されてから、LMSInitialize (“ ”) API 関数を呼び出すまでの状態に相当する。この状態において、SCO は、LMS によって提供されてる API アダプタを発見しなくてはならない。いったん、API アダプタが見つければ、SCO は、以下の API 関数を呼び出すことができる。

- LMSInitialize(“”)
- LMSGetLastError()
- LMSGetErrorString()
- LMSGetDiagnostic()

Initialized(初期化済み): 初期化済み: これは、SCO が、LMSInitialize (“ ”) API 関数を呼び出してから、LMSFinish(“ ”) API 関数を呼び出すまでの状態である。SCO が初期化済み状態にあるならば、LMSInitialize(“ ”)を除く全ての API 関数呼び出しが許される。

Finished(完了): これは、SCO が、LMSFinish(“ ”) API 関数を呼び出した状態である。もし、API アダプタが LMSFinish(“ ”)を呼び出した SCO に” false “を返した場合、SCO は以下の API 関数を呼び出すことができる。

- LMSGetLastError()
- LMSGetErrorString()
- LMSGetDiagnostic()

もし LMS API アダプタが “false” を返せば、LMS API アダプタがいかなる API 関数呼び出しにも適切に応答する保証はない。

3.3.3. API エラー・コードの使用法

SCO は、すべての API 呼び出しが成功したか否か、またもし不成功の場合、何が悪かったかを評価する方法を持たねばならない。LMSGetLastError()関数は、前の API 関数呼び出しによってもたらされたエラーのタイプを決定するために使用することができるエラーコードを返す。

構文: LMSGetLastError()

パラメータ: なし

返り値型: 文字列- 値は次のテーブル中で識別される整数番号に変換することができる。

コード	記述	使用法
"0"	No Error(エラーなし)	エラーなし。API 呼び出し成功。
"101"	General Exception(一般的な例外)	常に一般的な例外を示すために使用する。
"201"	Invalid argument error(無効な引き数エラー)	存在しない SCORM 実行環境データモデル要素への呼び出しが行なわれたとき使用される。 不正な引数が API 経由で引き渡されたとき使用される。
	例えば、LMSGetValue("cmi.core.zip_code") "cmi.core.zip_code"は有効な CMI データモデル要素ではない。 例えば、LMSInitialize("init") LMSInitialize("")は、空文字列である。	
"202"	Element cannot have children(要素は子を持つことができない)	LMSGetValue()が、_children をサポートしていない、いずれかのデータモデルカテゴリあるいは要素に対して呼ばれたときに使用される。
	例. LMSGetValue("cmi.student_id_children")	
"203"	Element not an array(配列でない要素である)	LMSGetValue()が、_count をサポートしていない、いずれかのデータモデルカテゴリあるいは要素に対して呼ばれたときに使用される。
	Cannot have count(数値を持たない)	

	例. <i>LMSGetValue</i> (“ <i>cmi.core_count</i> ”)	
“301”	Not Initialized(未初期化)	<i>LMSInitialize</i> (“”)が呼ばれる前に、いずれかの API 関数が呼び出された場合に使用される。
“401”	Not implemented error(非実装エラー)	LMS がサポートしていないデータモデル要素を呼び出したとき、もしくは、SCORM 実行環境データモデル以外のデータモデルが使われた場合に使用される。
	例. <i>LMSGetValue</i> (“ <i>cmi.objectives.0.id</i> ”) “ <i>cmi.objectives.0.id</i> ” 要素はオプション要素である。LMS がこの要素をサポートしない場合、LMS は空文字列(“”)を返し、エラーコードを 401 に設定しなくてはならない。 例. <i>LMSGetValue</i> (“ <i>xyz.score.result</i> ”) “ <i>xyz.score.result</i> ” は、有効な SCORM 実行環境データモデル要素ではない。それが、有効なデータモデル要素(“ <i>cmi</i> ”)に対して“ <i>xyz</i> ”)ではないので、LMS はエラーコード 401 を設定し、空文字列を返さなくてはならない。	
“402”	Invalid set value,element is a keyword(無効なセット値であり、要素がキーワード)	<i>LMSSetValue</i> ()呼び出しがキーワードに対して行われた場合に使用される。
	例. <i>LMSSetValue</i> (“ <i>cmi.core._children</i> ” , “ <i>student_id.student_name</i> ”) “ <i>cmi.core._children</i> ” はキーワード。	
“403”	Element is read only(要素は読み出し専用)	<i>LMSSetValue</i> ()呼び出しが、読み出し専用要素に対して行われた場合に使用される。
	例. <i>LMSSetValue</i> (“ <i>cmi.core.student_id</i> ” ” <i>JoeStudent</i> ”) “ <i>cmi.core.student_id</i> ” は読み出し専用の CMI 要素。	
“404”	Element is write only(要素は書き込み専用)	<i>LMSGetValue</i> ()呼び出しが、書き込み専用要素に対して行われた場合に使用される。
	例. <i>LMSGetValue</i> (“ <i>cmi.core.exit</i> ”) “ <i>cmi.core.exit</i> ” は書き込み専用の CMI 要素。	
“405”	Incorrect Data Tyoe(不正データタイプ)	要素に不正なデータタイプを設定しようとした場合に使用される。

	<p>例. <i>LMSSetValue("cmi.core.score.raw","eighty five")</i> " eighty five "は”cmi.core.score.raw”の正しいデータ型ではない。 正しいデータ型は、<i>CMIDecimal("85")</i>または <i>CMIBlack("")</i>である必要がある。</p> <p>例. <i>LMSSetValue("cmi.core.lesson_status","Not Attempted")</i> "Not Attempted"は、cmi.core.lesson_status 要素の有効なボキャブラリメンバではない。 ボキャブラリは、大文字と小文字の区別ができて、さらに完全に合致しなければならない。</p>
--	---

3.3.4. API の一般規則

次のリストは、API の一般的な使用法規則の要約である。

- 関数名はすべて大文字と小文字の区別ができて、正確に表わされなければならない。
- 関数パラメータまたは引数は、大文字と小文字の区別ができ、全てのパラメータは、小文字である。
- API 関数への各呼び出しは、エラーハンドリング関数を除いて、エラーコードをリセットする。

3.3.5. LMS の責任範囲

3.3.5.1. API アダプタ

LMS は、前の節に記述された API 要求機能を実装する API アダプタを提供しなくてはならない。SCO が下位の通信インフラストラクチャーについて知る必要がなく、代わりに、標準化された LMS API の存在のみに完全に依存すればよいように、このアダプタは特別のアダプタ実装の詳細を SCO に隠蔽する必要がある。API アダプタを使用するための必要条件は以下のとおりである。

- LMS は、API アダプタを含んでいる子ウィンドウあるいは LMS ウィンドウの子フレームであるブラウザ・ウィンドウで SCO を起動しなければならない。
- LMS は API アダプタを提供しなければならない。
- SCO から API インタラクションのためにサポートされる唯一のメカニズムは、ECMAScript(Java スクリプト)呼び出しを経由する。
- API アダプタは“API”と命名されるオブジェクトとして DOM を経由してアクセス可能でなければならない。

例として、API アダプタは、以下のような署名を持つ Java アプレットとして実装されるであろう。

```
public class API extends Applet
{
    public String LMSInitialize( String parameter )
    { . . . }
    public String LMSGetValue( String element )
    { . . . }
    public String LMSSetValue( String element, String value )
    { . . . }
    public String LMSCommit( String parameter )
    { . . . }
    public String LMSFinish( String parameter )
    { . . . }
    public String LMSGetLastError()
    { . . . }
    public String LMSGetErrorString( String errorCode )
    { . . . }
    public String LMSGetDiagnostic( String errorCode )
    { . . . }
}
```

例えば、C++のような他のプログラミング言語で API アダプタを実装することができ、例えばブラウザプラグインとして読み込むことができることに注意されたい。API アダプタの実装は LMS 固有であってよく、上記のコードは単なるアプローチ例に過ぎない。LMS API アダプタを実装するには多くの方法がある。

3.3.6. SCO の責任範囲

3.3.6.1. Find API

SCO は、少なくとも、LMSInitialize(“ ”)および LMSFinish(“ ”)API を呼び出さなくてはならない。そのためには、コンテンツは LMS によって提供される API アダプタを見つけなければならない。LMS は、API アダプタを DOM ウィンドウ階層に置いて、SCO が API を見つけるために再帰的に親およびオープンウィンドウ階層を捜すことができるようにしなくてはならない。コンテンツは、LMS API アダプタを見つけてコミュニケーションを確立しなくてはならない。SCO がこれを実行するためにどのような方法を選択をするかは、SCORM によって決められていない。

次のコード例は、LMS API アダプタを見つけるために SCO が使用することができるアルゴリズムを示す。これらの関数の使用は SCORM によって課された要求

ではなく、また、SCO は LMS API アダプタを見つけるために他のアプローチを使用することができる.

```
<SCRIPT LANGUAGE=JAVASCRIPT >
var findAPITries = 0;

function findAPI(win)
{
    // Check to see if the window (win) contains the API
    // if the window (win) does not contain the API and
    // the window (win) has a parent window and the parent window
    // is not the same as the window (win)
    while ( (win.API == null) &&
            (win.parent != null) &&
            (win.parent != win) )
    {
        // increment the number of findAPITries
        findAPITries++;

        // Note: 7 is an arbitrary number, but should be more than sufficient
        if (findAPITries > 7)
        {
            alert("Error finding API -- too deeply nested.");
            return null;
        }

        // set the variable that represents the window being
        // being searched to be the parent of the current window
        // then search for the API again
        win = win.parent;
    }
    return win.API;
}

function getAPI()
{
    // start by looking for the API in the current window
    var theAPI = findAPI(window);

    // if the API is null (could not be found in the current window)
    // and the current window has an opener window
    if ( (theAPI == null) &&
        (window.opener != null) &&
        (typeof(window.opener) != "undefined") )
    {
        // try to find the API in the current window's opener
        theAPI = findAPI(window.opener);
    }
    // if the API has not been found
    if (theAPI == null)
    {
        // Alert the user that the API Adapter could not be found
        alert("Unable to find an API adapter");
    }
    return theAPI;
}
```

3.4. データモデル

3.4.1. データモデルの概要

共通のデータモデルを確立する目的は、異なる LMS 環境によっても、SCO に関する定義された情報セットを確実に追跡することができるようにすることである。例えば、学習者の得点（スコア）追跡が一般的な要求であることとした場合、コンテンツは LMS 環境に得点を報告するための共通の方法を確立することが必要である。もし SCO が彼ら自身の特有な得点表現を使用するならば、学習管理システム(LMS)はその情報をどのように受取り、格納し、処理するかが分からない。

様々なコミュニティおよび標準組織の中で多くのデータモデルが開発中である。これらのデータモデル仕様書案は、SCO と LMS 環境の間で交換される情報セットを機能的にグループ分けしようとしている。例えば、以下のようなものである。学習者プロフィール情報、質問およびテストのインタラクション(相互作用)、状態情報、評価など。SCORM のこのバージョンの公表時点では、これらのデータモデル仕様書案は、まだ開発中であり、広く実装されていないし、テストもされていない。

3.4.1.1. SCORM ランタイム環境データモデル

このセクションのデータモデルは、SCORM ランタイム環境データモデルとして定義され、AICC CMI Guidelines for Interoperability に記述された AICC CMI データモデルから直接導かれたものである。AICC CMI データモデルは、よく定義されており、過去に実装が行われているので、SCORM に取り入れるよう選択された。将来、新しいデータモデルセットが採用され SCORM に組み込まれることが期待される。新しいデータモデルセットが定義され採用された場合、SCORM のこのバージョンの中で定義されたデータモデル要素は、そこにマッピングされることになる。現在のデータモデルから新しい要素へのマッピングは、将来のデータモデルへの比較的スムーズな移行パスを提供しなければならない。

使用するデータモデルを識別するため、このセクションで記述される要素の名前は全て"cmi"から始まっている。これは実装者に対して、これらの要素が AICC CMI データモデルの一部であることを示す。他のデータモデルが開発されれば、異なる指定(例えば cmi.elementName の代わりに adl.elementName)で始められるであろう。

SCORM のテストおよび評価過程に、AICC および IEEE のメンバは、AICC CMI データモデル中の要素の数を本質的に縮小することを決定した。これは開発中の新しいデータモデルへの移行を容易にすることと、削除された多くの要素

は実際には多くの実装者が実装していなかったからという二つの理由から行われた。SCORM 1.0 版に現われた AICC CMI データモデルから取り除かれた要素のリストは付録 C の中に要約されている。ここに含まれている SCORM ランタイム環境データモデルは、縮小された AICC CMI データモデルセットと一致している。

3.4.1.2. SCORM ランタイム環境データモデルの一般規則

次のリストは、データモデルのための一般的な使用規則を要約したものである。

- データ要素名中の第 1 シンボルはデータモデルを示す。例えば、”cmi”は AICC CMI データモデルを示す。これは、同じ API が他のデータモデルと共に使用することができることで API の機能性を拡張する。
- 3 つの予約キーワードがある。これらはすべて小文字で、下線(アンダースコア)から始まる。
 - ・ `_version`:キーワードは、LMS によってサポートされるデータモデルのバージョンを決めるために使用される。
 - ・ `_children`:キーワードは、どのデータモデル要素が LMS によってサポートされるかを定めるために使用される。
 - ・ `_count`:キーワードは、現リスト中の要素の数を定めるために使用される。
- 全ての配列は 0 が基底の配列である。アイテムは連続した形式で配列中に置かれなければならない。
- データモデル名は、大文字と小文字が区別される。
- データモデルは、一つの SCO の上に SCO の基準で実装される。一つの SCO は、もう一つの SCO のデータ・エレメントにアクセスすることができない。

3.4.2. データモデル要素

データモデル要素は、必須と任意の 2 つのカテゴリへ分割される。AICC CMI001 ガイドラインの相互運用性ドキュメントは、どの要素が LMS による必須な実装を要求するか、そしてどれが任意かを規定している。

LMS は必須データモデル要素をすべてサポートしなければならない。LMS 環境は、オプションデータモデル要素のすべてあるいは一部のサポートを実装することができる。

すべてのデータ要素の SCO からの利用は任意である。SCO は API 関数の `LMSInitialize(“”)` および `LMSFinish(“”)` を使用しなくてもよいが、`LMSSetValue()` や `LMSGetValue()` は使用しなくてもよい。SCO は非常に小さく、詳細を追跡しないようには設計されていてもよい。しかし、追跡される場合は、複数の LMS 環境にまたがって再利用されるよう、共通データモデルに適合していなければならない。

3.4.3. リストの扱い

リストないし配列に現われるいくつかのデータ要素がある。この一例は “Objective (学習目標)” である。コンテンツの中には複数の “学習目標” があるかもしれない、そして、学習者は “学習目標” を何回も経験してよい。

リストの値を取得・設定するために、インデックス値が使用される。インデックス値が省略できるのはリスト中にメンバが 1 つだけの場合のみである。インデックス番号付けは 0 からスタートする。値がリストに追加される場合、SCO は使用されている最後のインデックス値を知らなければならない。

新しい配列要素はすべて連続して加えられるものとする。SCO は配列番号をスキップしてはならず、配列値のリストを作成するとき、空の配列要素を残してはならない。_count キーワードは、リスト中のレコード数を決定するために使われる。たとえば、現在格納されている対象レコードの数を決定するために、以下の API 呼び出しが使われる。

```
var numOfObjectives = LMSGetValue("cmi.objective._count");
```

学習目的レコードの数が分からない場合、SCO が現在の数を 0 から開始してよい。これは、第 1 のインデックス位置に現在格納された学習目標に関して任意の情報に上書きできるということである。上書きするか、追加するかは、SCO 作成者によって開発時に決定される。

リストの中の要素は、点-番号表記法 (.n によって表される) で参照される。たとえば、SCO の最初の学習目標の状態要素の値は、`"cmi.objective.0.status"` で参照される。第 4 番目の学習目標の状態要素は、`"cmi.objective.3.status"` で参照される。

3.4.4. SCORM ランタイム環境データモデル

cmi.core	
<p>すべての LMS システムによって提供されることが必要な情報。すべての SCO が起動時に必要とする可能性があるもの。</p> <p>cmi.core children:</p> <p>student_id, student_name, lesson_location, credit, lesson_status, entry, score, total_time, lesson_mode, exit, session_time</p>	
cmi.core._children	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: _children キーワードは、LMS でサポートされる core カテゴリの要素を全て決定するために使用される。要素が、サポートはされているが、子を持たなければ、空文字列が返される。要素がサポートされていないならば、空文字列が返される。次に最後のエラーを要求すれば、要素がサポートされないことを確認できる。</p> <p>使用法: どの cmi.core データ要素が LMS でサポートされるかを決定する。</p> <p>形式: 返り値は、LMS でサポートされる core カテゴリの要素名全てをコンマで区切ったリストである。</p> <p>LMS の動作:</p> <ul style="list-style-type: none">・ 初期化: このグループでサポートされた子要素の組。LMSGetValue()に対して、LMS でサポートされた子要素の適切なリストが返るようにする。・ LMSGetValue(): LMS は、サポートされた要素のリストをコンマで区切って返す<ul style="list-style-type: none">・ API 呼び出しの例: LMSGetValue("cmi.core._children")・ 返り値の例: "student_id,student_name,lesson_location,credit,lesson_status,entry,score,total_time,exit,session_time"・ エラーコード:<ul style="list-style-type: none">401 非実装エラー cmi.core._children 要素がサポートされていないならば、空文字列が返され、そして要素がサポートされないことを示すエラーコードが設定される。 <i>注意:</i> この要素は必須なので、cmi.core._children 要素は LMS によってサポートされなければならない。・ LMSSetValue(): LMS は、以下に従ってエラーコードを設定しなければならない。<ul style="list-style-type: none">・ エラーコード:<ul style="list-style-type: none">402 無効なセット値, 要素はキーワードである。 もし、この要素がサポートされ（この要素は必須なので、LMS でサポートされなければならない）、そし

	<p>て、要求側がこの要素で <code>LMSSetValue()</code> を呼び出そうとするならば、LMS はエラーコードを 402 に設定しなければならない。</p> <p>401 — 非実装エラー この要素がサポートされない場合、サポートされないことを示すために LMS によってエラーコード 401 が設定される。</p> <p><i>注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。</i></p> <p>SCO の使用例: SCOs は、特定の要素が LMS によって実装されているかどうか決定するために <code>cmi.core._children</code> 要求を使うことができる:</p> <pre>var coreChildren = LMSGetValue("cmi.core._children"); if (coreChildren.indexOf("student_name") != -1) { studentName=LMSGetValue("cmi.core.student_name"); }</pre>
cmi.core.student_id	
<p>Supported API calls: <code>LMSGetValue()</code></p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMIIentifier</p> <p>SCO のアクセス可能性: 読み取りのみ</p>	<p>定義: LMS システムのひとりのユーザを参照するユニークな英数字または識別子。</p> <p>使用法: 学習者をユニークに識別するために使用。</p> <p>形式: スペースなしで最大 255 文字の英数字。ダッシュ(またはハイフン)とアンダーラインは、含んでよい。ピリオドは、許されない。大文字、小文字は区別しない。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 学習者登録に基づいて、LMS が実行しなくてはならない。 ・ LMSGetValue(): 学習者に対し、LMS によって保存される現在の値を返す。 <ul style="list-style-type: none"> ・ 戻り値の例: “Joe_Student1” “JS-2000” “joe2000-3” ・ エラーコード: 401 — 非実装エラー. この要素がサポートされない場合、空文字列は返され、この要素がサポートされないことを示すエラー・コードが設定される。 <i>注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。</i> ・ LMSSetValue(): LMS は下記に従ってエラーコードを設定しなくてはならない。

	<p>・エラーコード：</p> <p>403 — 要素は読み出し専用. この要素がサポートされ(この要素は必須であるので, LMS はこの要素をサポートしなければならない), この要素に対して LMSSetValue()の呼び出しが行われる場合, LMS はエラーコードを 403 に設定しなくてはならない.</p> <p>401 — 非実装エラー. この要素がサポートされない場合, LMS はこの要素がサポートされないことを示すために, エラーコードは 401 に設定される. 注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</p> <p>SCO の使用例: 起動された場合, SCO は学習者 ID を表示しようとする.</p> <pre>var coreStudentID = LMSGetValue("cmi.core.student_id");</pre>
cmi.core.student_name	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory:Yes</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み取りのみ</p>	<p>定義: 通常, コース名簿上の学習者の公式名. ファーストネームではなくフルネーム.</p> <p>使用法: 学習者の公式な氏名を表すために使用される.</p> <p>形式: 姓, 名およびミドルネームのイニシャル. 姓と名はコンマで区切られる. 名の中の空白はまもられなくてはならない.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 学習者登録に基づいて, LMS が実行しなくてはならない. ・ LMSGetValue(): LMS によって格納された現在値を返す. <ul style="list-style-type: none"> ・ 返り値の例: <pre>"Student, Joseph A." "Student, Mike A. Jr."</pre> ・ エラーコード: <p>401 — 非実装エラー. この要素がサポートされない場合, 空文字列を返し, この要素がサポートされていないことを示すためにエラーコードが設定される. 注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</p> ・ LMSSetValue(): LMS は下記に従ってエラーコードを設定しなくてはならない. : <ul style="list-style-type: none"> ・ エラーコード: <p>403 — 要素は読み出し専用. この要素がサポートされ(この要素は必須であるので, LMS はこの要素をサポート</p>

	<p>ートしなければならない), この要素に対して LMSSetValue()を起動しようとする場合, LMS はエラーコードを 403 に設定しなくてはならない.</p> <p>401 — 非実装エラー. この要素がサポートされない場合, LMS はこの要素がサポートされないことを示すために, エラーコードを 401 に設定する.</p> <p><i>注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</i></p> <p>SCO の使用例: SCO は, SCO の起動で学習者名を表示しようとする.</p> <pre>var coreStudentName = LMSGetValue("cmi.core.student_name");</pre>
cmi.core.lesson_location	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory :Yes</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: これは, 学習者が SCO を体験した最後に, LMS システムに渡される SCO の出口ポイントに相当する. これは, 学習者に以前に出た SCO の同じ位置に戻らせるためのメカニズムを提供するものである. 言い換えれば, この要素は学習者の出口ポイントを特定することができ, また, 次に学習者が SCO を実行する時, SCO はその出口ポイントを入口ポイントとして使用することができる.</p> <p>使用法: この要素は, 学習者が最後に SCO 内にいた場所を定義する. この要素は, セッション中に「ブックマーク」(しおり)を記憶しておくために SCO によって使用される. SCO が中断され, その後再入する場合, 学習者が退出した所に, 学習者を送り返すために SCO は lesson_location を使用することができる.</p> <p>形式: 実装に依存. LMS システムは単にこのデータを保持し, SCO が求めれば, 学習者が再び入る場合, SCO にそれを返す. SCO が LMS システムに渡すものすべてが返される. フォーマットは, SCO が期待するものすべてと一致する--フォーマットは SCO によって作成される. 学習者が最初に SCO に入る場合, あるいは, 好ましい開始点がない場合, lesson_location は空文字列(“”)となる.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は空文字列にこれを設定しなければならない. SCO は, 任意にこの値を設定し, SCO への再入する際, そのデータ取り出す. • LMSGetValue(): LMS によって格納された現在値を返す. <ul style="list-style-type: none"> • エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされないことを示すために, エラーコードを(401に)設定する.</p> <p><i>注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</i></p> • LMSSetValue(): 与えられた値をデータモデル要素に設定す

	<p>る。値は、この要素のデータ・タイプと一致しなければならない。</p> <ul style="list-style-type: none"> ・エラーコード： <ul style="list-style-type: none"> 405 — 不正データ・タイプ。 この要素がサポートされ (この要素は必須であるので、LMS はこの要素をサポートしなければならない)、正しくないデータ・タイプで LMSSetValue() を起動しようとした場合。 401 — 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされないことを示すために、エラーコードを(401 に)設定する。 <i>注意：この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。</i> ・ 返り値/設定値の例： SCO の実装に依存。 <p>SCO の使用例:</p> <p>SCO は「しおり」として lesson_location (前回実行位置) を使用することができる。</p> <ul style="list-style-type: none"> ・ SCO の起動時には、学習者が以前の試みで SCO から退出した位置に、SCO は位置決めする。 ・ SCO からの退出時には、学習者が SCO に再入する時に、その位置が検索できるように、位置決めする。 <pre>// Example SCO that is written using JavaScript and // HTML with anchors // This function could exist in a function call onLoad() that gets invoked // when the HTML page is loaded Var coreSCOLocation = LMSGetValue("cmi.core.lesson_location"); if (LMSGetLastError() == "0") { // coreSCOLocation contains an anchor name // defined in the HTML page // Start the SCO off where the student left off window.location.hash = coreSCOLocation; } else { // Error processing }</pre>
cmi.core.credit	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データタイプ: CMIScore (Number)</p>	<p>定義: この SCO の実行 (合格/不合格、得点) に基づき、学習者の履修記録が LMS システムによって更新されるかどうかを示す。</p> <p>使用法: LMS システムによって、学習者が履修記録のために SCO を実行しているかどうかを示すために使用される。</p>

<p>CMIVocabulary(Credit) “credit” “no-credit”</p> <p>SCO のアクセス可能性: 読み取りのみ</p>	<p>cmi.core.credit は、lesson_mode と組み合わせて使用される。詳細は、cmi.core.lesson_mode を参照。cmi.core.credit は、また lesson_status の決定にも使用される。詳細は、cmi.core.lesson_status を参照。</p> <p>形式: ボキャブラリの集合。2つの可能なボキャブラリ値:</p> <ul style="list-style-type: none"> “credit”: 学習者が SCO を履修記録のために実行していることを意味する。LMS システムは、SCO が LMS システムにデータを送信する場合、LMS システムがそれを学習者に対する履修記録とすることを SCO に示す。 “no credit”: 学習者が SCO 履修記録のためでなく実行していることを意味する。彼の現在の履修記録は、どのようなものであれ（たとえば 80 点で合格状態）、この SCO における彼の成績によっては変更されない。LMS システムは、SCO が LMS システムにデータを送ったとしても、それが学習者の単位認定を変更しないことを、SCO に伝える。 <p>LMS の動作:</p> <ul style="list-style-type: none"> 初期化: LMS は、学習者が履修記録のために SCO を実行しているかどうかを決定しなくてはならない。注意: 現在、SCORM には、学習コンテンツがクレジットかノークレジットかを意味付けるための決められた方法がない。したがって、実装は LMS 特有(依存)である。 LMSGetValue(): LMS によって格納された現在値を返す。 <ul style="list-style-type: none"> 返り値の例: "no credit" "credit" エラーコード: 401 — 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされないことを示すために、エラーコードを(401 に)設定する。注意: この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 LMSSetValue(): LMS は下記に従って、エラーコードを設定しなければならない <ul style="list-style-type: none"> エラーコード: 403 — 要素は読み出し専用。 この要素がサポートされ(この要素は必須であるので、LMS はこの要素をサポートしなければならない)、この要素に対して LMSSetValue()を起動しようとする場合、LMS はエラーコードを 403 に設定しなくてはならない。 401 — 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされないことを示すために、エラーコードを(401 に)設定する。注意: この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 <p>SCO の使用例: SCO は、ブラウザに何を表示するか決めるために LMS から返された値を使用する。Credit か no-credit かによって、SCO の中に異なる表</p>
--	--

	<p>示コンテンツがあってもよい.</p> <pre> var creditFlag = LMSGetValue("cmi.core.credit") if (creditFlag == "credit") { // Student is taking course for credit handle appropriately } else { // Student is taking course for no credit, handle appropriately } </pre>
cmi.core.lesson_status	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMIVocabulary(Status)</p> <p>passed completed failed incomplete browsed not attempted</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: これは LMS システムによって決定される現在の学習者ステータスである. 6つのステータス値をとれる.</p> <p>使用法: 通常, SCO はそれ自身のステータスを決定し, それを LMS へ渡す.</p> <ol style="list-style-type: none"> もし cmi.core.credit が「クレジット」に設定され, マニフェスト(教材定義)の中に合格点(adlcp:masteryscore)がある場合, LMS は, 合格点と比較して, 学習者のスコア次第で, 合格か不合格かへステータスを変更することができる. マニフェストの中に合格点(adlcp:masteryscore)がない場合, LMS は SCO が決定したステータスを上書き (優先) することはできない. 学習者が SCO に対し「ノークレジット」とするならば, 1つの例外を除き, lesson_status に変化はない. Lesson_mode が「閲覧」(browse)である場合は, たとえ cmi.core.credit が「ノークレジット」に設定されても, lesson_status は「閲覧された」に変更されるであろう. <p>SCO への再入では, LMS はステータスを, 「パス (合格)」か, 「失敗した(不合格)」か, 「閲覧済み」に変更する. SCO の (マニフェストに定義された) 合格基準に従い, パス(合格)か失敗 (不合格)となる. SCO が「閲覧」の lesson_mode で最初の起動が行われたとき, 「閲覧済み」となる.</p> <p>形式: ボキャブラリの集合. 6つの可能なボキャブラリ値.</p> <ul style="list-style-type: none"> 合格(passed): SCO での必要な数の学習酷評が習得された, あるいは, 必要な得点が達成された. 学習者は, SCO を完了して, 合格したと見なされる. 完了(complete): SCO は合格か, 不合格かわからないが, SCO の全ての要素は学習者によって経験された. 学習者は, SCO を完了したとみなされる. たとえば, 「合格」は LMS システムに知られている特定の得点(スコア)に依存する. SCO は素点を

	<p>知っているが、その素点が「合格」に十分な高いスコアかどうかは知らない.</p> <ul style="list-style-type: none"> ・ 不合格(failed) : SCO は合格しなかった. 全ての SCO 要素は学習者によって完了したか, 完了しなかったかである. 学習者は SCO を完了したが, 不合格とみなされる. ・ 非完了(incomplete):SCO は開始されたが, 終了されなかった. ・ 閲覧済み (browsed) : 学習者は, 最初の試みで, 閲覧の LMS モードで SCO を起動した. ・ 未試行(not attempted): 非完了(incomplete)は, 学習者が SCO を実行する試みをするが, 何かの理由でそれを終えることができなかったことを意味する. 未試行(not attempted)は, 学習者が SCO を開始さえしなかったことを意味する. 多分, 学習者はちょっと目次または SCO 要約を読んで, 準備ができていないと判断したのだろう. SCO 内のいくつかのアルゴリズムは, SCO が未試行(not attempted)から非完了(incomplete)へ SCO がいつ移るかを決定するために使われる. <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 学習者が SCO への最初の試行の場合, lesson_status は, 未試行(not attempted)に設定される. LMS は, 初期値を未試行(not attempted)に設定しなくてはならない. <ul style="list-style-type: none"> ・ LMS 動作への追加要求: SCO が cmi.core.lesson_status を設定すれば, 何の問題もない. しかしながら, SCORM は, SCO に cmi.core.lesson_status を設定するようには強制していない. 以下のケースを正しく処理するために, 守らなくてはならない幾つかの追加要求がある. <ul style="list-style-type: none"> ・ 最初の LMS 起動では, lesson_status は, 未試行(not attempted)に設定すべきである. ・ LMSFinish () 呼び出しを受け取った, または利用者が SCO から離れた場合, LMS は SCO の cmi.core.lesson_status を完了(complete)に設定すべきである. ・ cmi.core.lesson_status を完了(completed)に設定した後は, LMS は合格点が cmi.student_data.mastery_score に指定されているかをチェックすべきである. もし, 合格点が提供されており, SCO が cmi.core.score.raw を設定済みなら, LMS は cmi.core.score.raw を合格点と比較し, cmi.core.lesson_status を「合格」または「不合格」に設定する. もし, 合格点が提供されていないなら, LMS は, cmi.core.lesson_status を「完了」のままにしておく. ・ LMSGetValue():データモデルに格納された値を返す. リターンは, ステータスのボキャブラリセット中の 1 つでなければならない. <ul style="list-style-type: none"> ・ エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされない
--	--

	<p>ならば、空文字列を返しそしてエラーコードはこの要素がサポートされていないことを示すために設定されなければならない。注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。</p> <ul style="list-style-type: none"> LMSSetValue(): 与えられた値をデータモデル要素に設定する。値は、この要素のデータ・タイプと一致しなければならない。 <ul style="list-style-type: none"> エラーコード： <ul style="list-style-type: none"> 405 – 不正データ・タイプ この要素がサポートされ(この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない)、不正データ・タイプの値で LMSSetValue()を呼び出そうとした。 401 – 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない。注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。 返り値/設定値の例: "completed" "failed" "browsed" <p>SCO 使用法:</p> <ul style="list-style-type: none"> passed – SCO が履修記録のために実行されている場合に使用される。 failed – SCO が履修記録のために実行されている場合に使用される。 completed – SCO が履修記録のためでなく実行されている場合に使用される。 incomplete – SCO が履修記録のために、あるいは履修記録のためでなく実行されている場合に使用される。SCO が (合格, 不合格, 完了のステータスが決定される前に), 早期に終了してしまった時に使用される。 browsed – lesson_mode が閲覧であるとき使用される。 not attempted – SCO は lesson_status を決して not attempted に設定してはならない。(学習者が最初に SCO を試行するとき LMS によって初期化される) <p>SCO の使用例:</p> <pre> var lessonStatus = LMSGetValue("cmi.core.lesson_status"); if (lessonStatus == "failed") { // Student failed the SCO, handle appropriately </pre>
--	---

	<pre> } else { // Student did not fail the SCO, handle appropriately } </pre>
cmi.core.entry	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMIVocabulary(Entry) "ab-initio" "resume" "" - empty string</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: 学習者が SCO に以前入ったことがあるかどうかの表示.</p> <p>使用法: 学習者が初めて SCO に入るとき, cmi.core.entry 要素は LMS によって ab-initio に設定されなければならない. 学習者が中断していた SCO に再入する場合, エントリフラグは LMS によって resume に設定されなければならない.</p> <p>形式: ボキャブラリの集合. 3つの可能なボキャブラリ値:</p> <ul style="list-style-type: none"> 「ab-initio」(初回試行):これは, 学習者が初めて SCO に入るときを示す. 学習者がプリテストを完了して SCO 中の学習目標の全てを合格しているかもしれないので, lesson_status が未試行(not attempted)であることは, 信頼できる指標ではない. すなわち, SCO は, 学習者が見なくても合格となるかもしれない. 「resume」(中断再開):これは, 学習者が以前に SCO に入ったことを示す. 学習者は, 中断していた SCO を再開している. 「」: 空文字列は, 初回実行(ab-initio)でも, 停止した状態(resume)からの継続でもなく, SCO に入ったことを表わすために使われる. これが使われるのは, SCO がすでに完了した後で, それが復習のためにロードされた場合である. この場合は, 最初の起動(ab-initio)でも中断状態からの再開(resume)でもない. <p>LMS の動作:</p> <ul style="list-style-type: none"> 初期化: SCO 最初の起動に際して, LMS は, データモデル値を「ab-initio」に初期化しなくてはならない. 追加動作: LMS が LMSFinish()を受け取るか, または利用者が SCO から離れた場合, LMS は cmi.core.entry に "" (空文字列) または "resume" を設定しなければならない. LMS は, SCO が cmi.core.exit に設定した値を調べることによってこれを決定する. SCO が cmi.core.exit の値に "suspend" を設定すれば, LMS は, 次の SCO 起動に備えて, cmi.core.entry に "resume" を設定する. もし, SCO が cmi.core.exit の値に "resume" 以外を設定するかまったく値を設定しない場合, LMS は cmi.core.entry に "" (空文字列) を設定する. LMSGetValue(): データモデルに格納された値を返す. 返り値は cmi.core.entry データ要素用にとって, 以下のボキャブラ

	<p>リセット中の 1 つでなければならない。</p> <ul style="list-style-type: none"> ・ 返り値の例 : "ab-initio" "resume" ・ エラーコード: 401 - 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. <i>注意: この要素は (サポート) 必須でないので, LMS はこれをサポートしなければならない.</i> ・ LMSSetValue(): LMS は下記に従って、エラーコードを設定しなければならない. ・ エラーコード: 403 - 要素は読み出し専用. この要素に対して、LMSSetValue() を起動する要求があった場合、LMS はエラーコードを 403 に設定すべきである. 401 - 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない. <i>注意: この要素は (サポート) 必須でないので, LMS はこれをサポートしなければならない.</i> <p>SCO の使用例:</p> <pre> var entryStatus = LMSGetValue("cmi.core.entry") if (LMSGetLastError() == "0") { if (entryStatus == "resume") { // Student is resuming SCO } else { // This is the first time the student has entered the SCO } } else { // Error condition, handle appropriately } </pre>
cmi.core.score 学習者の得点表示 Children of cmi.core.score: raw, min, max	
cmi.core.score._children	

<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: _children キーワードは LMS によってサポートされるスコア・カテゴリのすべての要素を決定するために使用される。要素に子(no children)がないが、要素はサポートされる場合、空文字列が返される。この要素がサポートされない場合、戻り値はない。次に、最後のエラーを要求すれば、要素がサポートされないことを確認できる。</p> <p>使用法: LMS が cmi.core.score のどの子要素をサポートしているかを決定するために使用される。Raw はサポートされなければならない唯一の必須要素である。</p> <p>形式: 戻り値は LMS によってサポートされるスコア・カテゴリすべての要素名がコンマで区切られたリストである。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: このグループのためにサポートされた子のセット。その結果、LMSGetValue()リクエストでは、サポートされた子の適切なリストが返される。 • LMSGetValue(): サポートされた要素がコンマで区切られたリストを返す。 <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.core.score._children.") • 戻り値の例: "raw" - conformant LMS は少なくとも次の要素をサポートしなければならない。 "raw,min,max", "raw,min" • エラーコード: <ul style="list-style-type: none"> 401 - 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。注意: この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 • LMSSetValue(): LMS は下記に従って、エラーコードを設定しなければならない。 <ul style="list-style-type: none"> • エラーコード: <ul style="list-style-type: none"> 402 - 無効のセット値, 要素はキーワード. この要素に対して LMSSetValue()起動が試みられる場合、LMS はエラーコードを 402 に設定すべきである。 401 - 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない。注意: この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 <p>SCO の使用例:</p> <pre>var scoreChildren = LMSGetValue("cmi.core.score._children"); if (scoreChildren.indexOf("min") != -1) { LMSSetValue("cmi.core.score.min","10"); }</pre>
---	---

cmi.core.score.raw	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMIDecimal または CMIBlank</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: SCO に対しての最終試行における学習者の得点表示. この得点は, SCO 設計者へ意味のある方法で決定され, 計算される. 例えば, それは, 完了した学習目標の割合を反映できるし, 多項目選択のテスト中における未加工の点数でありえるし, あるいは, SCO の中に埋め込まれた質問に対する最初の正確なレスポンスの数を示すことができる.</p> <p>cmi.core.score.raw は, 0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習者が SCO に対しての最初の試行である場合, cmi.core.score.raw は""(空文字列)に設定されなければならない. 追加の試行については, cmi.core.score.raw が, 学習者の直前に行われた最後の試行で記録されたものを反映する. もし cmi.core.score.raw が SCO の中に設定されず, cmi.core.score.raw の要請が SCO によってなされたならば, 空文字列が返されなければならない("").</p> <p>形式: 10 進数あるいはブランク.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は, SCO の最初の起動で, これを空文字列("")に初期化しなければならない. SCO は, この値を設定する責任範囲がある. SCO がこの値を設定する前に LMSGetValue () が要請された場合, LMS は空文字列を返されなければならない(""). • LMSGetValue(): データモデルに格納された値を返す. 返された値は, CMIDecimal あるいは CMIBlank の型でなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("mi.core.score.raw") • エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. 注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</p> • LMSSetValue(): データモデル要素に与えられた値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: <p>LMSSetValue("cmi.core.score.raw" , "85.7")</p> • エラーコード: <p>405 — 不正データ・タイプ. この要素がサポートされ (この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない) , リクエストが,</p>

	<p>正しいデータ型ではない値で <code>LMSSetValue()</code> を起動しようとする場合.</p> <p>401 – 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない. <i>注意: この要素は (サポート) 必須でないので, LMS はこれをサポートしなければならない.</i></p> <ul style="list-style-type: none"> • 返り値/設定値の例: "90" "85.7" "" <p>SCO の使用例:</p> <p>SCO は学習者の SCO の中での素点 (得点) を追跡するためにこれを使用できる.</p> <p><code>LMSSetValue("cmi.core.score.raw","85")</code></p>
cmi.core.score.max	
<p>Supported API calls:</p> <p><code>LMGetValue()</code></p> <p><code>LMSSetValue()</code></p> <p>LMS Mandatory: No</p> <p>データ・タイプ:</p> <p>CMIDecimal または CMIBlank</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: 学習者が達成する可能性があった最高点あるいは総数.</p> <p><code>cmi.core.score.max</code> は, 0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習者が達成する可能性があった最高得点の表示.</p> <p>フォーマット: 10 進数あるいは空白.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は SCO の最初の起動でこれを空文字列("")に初期化しなくてはならない. SCO はこの値を設定する責任範囲がある. SCO がこの値を設定する前に, <code>LMGetValue()</code> が要求される (呼ばれる) 場合, LMS は空文字列("")を返すべきである. • LMGetValue(): データモデルに格納された値を返す. 返された値は, CMIDecimal あるいは CMIBlank の型でなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: <code>LMGetValue("cmi.core.score.max")</code> • エラーコード: <p>401 – 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> • LMSSetValue(): データモデル要素に与えられた値を設定する. 値は, この要素のデータ・タイプと合致しなければならない

	<p>ない.</p> <ul style="list-style-type: none"> • API 呼び出しの例: LMSSetValue("cmi.core.score.max","100") • エラーコード(Error Code): <ul style="list-style-type: none"> 405 –不正データ・タイプ. この要素がサポートされ、正しいデータ型ではない値で LMSSetValue() を起動しようとする場合. 401 – 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない. • 返り値/設定値の例: "100" "5" "" <p>SCO の使用例:</p> <pre>var scoreChildren = LMSGetValue("cmi.core.score._children"); if (scoreChildren.indexOf("max") != -1) { LMSSetValue("cmi.core.score.max","100"); }</pre>
cmi.core.score.min	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIDecimal または CMIBlank</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: 学習者が取得する可能性があった最低点</p> <p>cmi.core.score.min は、0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習者が取得する可能性があった最低得点の表示</p> <p>形式: 10 進数あるいは空白.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は SCO を最初に起動するとき、これを空文字列("")に初期化しなくてはならない. SCO はこの値を設定する責任範囲がある. SCO がこの値を設定する前に、LMSGetValue()が要求される場合、LMS は空文字列("")を返すべきである. • LMSGetValue(): データモデルに格納された値を返す. 返された値は、CMIDecimal あるいは CMIBlank の型でなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.core.score.min") • エラーコード: <ul style="list-style-type: none"> 401 – 非実装エラー. この要素がサポートされない

	<p>ならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。</p> <ul style="list-style-type: none"> • LMSSetValue(): データモデル要素に与えられた値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 • API 呼び出しの例: LMSSetValue("cmi.core.score.min","5") • エラーコード: <ul style="list-style-type: none"> 405- 不正データ・タイプ. この要素がサポートされ、正しいデータ型ではない値で LMSSetValue() を起動しようとする場合。 401 - 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない。 • 返り値/設定値の例: <pre> "10" "45.5" "" </pre> <p>SCO の使用例:</p> <pre> var scoreChildren = LMSGetValue("cmi.core.score._children"); if (scoreChildren.indexOf("min") != -1) { LMSSetValue("cmi.core.score.min","10"); } </pre>
cmi.core.total_time	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMITimespan</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: SCO での学習者のすべてのセッションの蓄積時間。</p> <p>使用法: ある学生のある SCO における全てのセッションで費やされる総計時間を追跡するために使用する。</p> <p>LMS は、SCO が最初に起動される時、cmi.core.total_time をデフォルト値に初期化し、その後、SCO の報告値 (session_time) を使用して総計時間を算出する。</p> <p>形式: 時間、分と秒は、コロンによって区分される。HHHH:MM:SS.SS 時間は、最低 2 桁で最高 4 桁を持つ。分はちょうど 2 桁から成る。秒は 2 桁の数字と、オプションで小数点と 1 桁あるいは 2 桁の追加の数字(つまり 34.45 のような)を含む。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は SCO の最初の起動で初期値を

	<p>「0000:00:00.00」にすべきである。</p> <p>追加動作：SCO は一回の実行中に、複数回 <code>cmi.core.session_time</code> を実行できる。SCO が <code>LMSFinish()</code> を発行するか、利用者が SCO から離れた場合、LMS は、SCO が設定した(一組あるとして)最終の <code>CMI.core.session_time</code> を取得して、その時間を <code>CMI.core.total_time</code> に積算しなければならない。SCO が引き続いて起動され、<code>CMI.core.total_time</code> について <code>LMSGetValue</code> が呼び出された場合、LMS は積算した時間を返さなければならない。LMS は、<code>CMI.core.session_time</code> についての <code>LMSGetValue</code> の呼び出しで、LMS に送られてくる複数の時間を積算してはならない。<code>CMI.core.session_time</code> について <code>LMSGetValue</code> の複数回呼び出しがあった場合、LMS は <code>cmi.core.session_time</code> として保持している値を上書きしなければならない。</p> <ul style="list-style-type: none"> • LMSGetValue(): データモデルに格納された値を返す。返された値は、<code>CMITimespan</code> の型でなければならない。 <ul style="list-style-type: none"> • API 呼び出しの例: <code>LMSGetValue(“cmi.core.total_time”)</code> • 返り値の例: <pre>“00:29:00” “01:27:45.5”</pre> • エラーコード: <ul style="list-style-type: none"> 401 ー 非実装エラー。 この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。 • LMSSetValue(): LMS は以下のようにエラーコードを設定しなくてはならない。 <ul style="list-style-type: none"> • エラーコード: <ul style="list-style-type: none"> 403 ー 要素は読み出し専用。 この要素に対して、<code>LMSSetValue()</code> を起動する要求があった場合、LMS はエラーコードを 403 に設定すべきである。 401 ー 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定しなければならない。注意：この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。 <p>SCO の使用例:</p> <pre>var totalTime = LMSGetValue(“cmi.core.total_time”); if (LMSGetLastError() == 0) { // Use cmi.core.total_time }</pre>
--	---

cmi.core.lesson_mode	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIVocabulary(Mode)</p> <p>"browse" "normal" "review"</p> <p>SCO のアクセス可能性: 読み取りのみ</p>	<p>定義: 起動後の SCO の望ましい動作を決定する。多くの SCO は単一の「動作」を有する。しかし、いくつかの SCO は異なる量の情報を表示したり、異なる順序で情報を表示したり、指導者や設計者の決定に基づいた異なるトレーニング哲学を反映する情報を表示することができる。設計者は、事実上無制限の多くの方法で SCO を動作させることができる。この標準は、異なる SCO の動作を生じさせる 3 つのパラメータの通信をサポートする。</p> <p>使用法: SCO が起動される異なるモードを表すために使用される。lesson_status と共に使用される。</p> <p>形式: ボキャブラリの集合。3 つの可能なボキャブラリの値は以下。</p> <ul style="list-style-type: none"> • “browse”: 学習者は、題材をプレビューしたいが、必ずしも評価のために SCO に挑戦しない。 • “normal”: これは、学習の単位認定を望む学習者のために設計された動作を示す。 • “review”: 学習者は、すでに少なくとも一回教材を見て評価された。 <p>認識不能かあるいは予期しない lesson_mode を受け取った場合、SCO は normal を受け取ったと仮定する。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は、SCO が起動されるモードを決定しなくてはならない。 注意: 現在、SCORM は学習コンテンツが異なるモードに入ることを通知する方法を定義していない。したがって、実装(方法)は LMS に依存する。 • LMSGetValue(): データモデルに格納された値を返す。返り(値)はエンタリー用セットボキャブラリのうちの 1 つでなければならない。 <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue(“cmi.core.lesson_mode”) • 返り値の例: “browse” “normal” “review” • エラーコード: 401 – 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためのエラーコードが設定される。 • LMSSetValue(): LMS は下記に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> • エラーコード:

	<p>403 — 要素は読み出し専用。この要素に対して、LMSSetValue()を起動する要求があった場合、LMS はエラーコードを 403 に設定すべきである。</p> <p>401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。</p> <p>SCO の使用例:</p> <pre> var mode = LMSGetValue("cmi.core.lesson_mode"); if (LMSGetLastError() == 0) { if (mode == "browse") { // Student is browsing the SCO } else if (mode == "review") { // Student has already seen an been graded // Must be reviewing material } else { // Student is launching the SCO in the normal mode } } </pre>
cmi.core.exit	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMIVocabulary(Exit)</p> <p>"time-out" "suspend" "logout" "" - empty string</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>定義: 学習者が SCO をどのようにそしてなぜ退出したかの表示。</p> <p>使用法: SCO から最後に退出した理由を示すために使用される。</p> <p>形式: ボキャブラリの集合。可能なボキャブラリ値は以下の 3 つ。</p> <ul style="list-style-type: none"> • “Time-Out”: これは、経過時間が長すぎると SCO が断定したか、もしくは max_time_allowed(許容試行時間)を超過したために、SCO が終了したことを示す。max_time_allowed は、マニフェストの中にある (このアイテムの adlcp:maxtimeallowed)。 • “Suspend”: これは、学習者が退出したポイントに、後で戻る意思があって SCO を退出することを示す。 • “logout”: これは、学習者が LMS へ戻ってログアウトするのではなく、学習者が内部からログアウトしたことを示す。これは、SCO がコントロールを LMS システムへ渡し、さらに、LMS システムは一適切なデータモデル要素を更新した後—自動的にコースから学習者をログアウトさせることを意味する。

	<ul style="list-style-type: none"> ・ “”: 空文字列は正常な退出状態を表わすために使用する. <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 要素は初期化される必要はない. LMSGetValue()(の API 呼び出し)はこの要素上で行われてはいない. 要素は SCO によってコントロールされる. <ul style="list-style-type: none"> ・ 追加動作: SCO は, cmi.core.exit に 4 つのうちの 1 つを設定する選択肢を持つ. その値に依存して, LMS 動作は以下のものでなければならない. <ul style="list-style-type: none"> ・ SCO が cmi.core.exit に “time-out” を設定した場合には, LMS は次の SCO 起動時に, cmi.core.entry に “” (空文字列) を設定しなければならない. ・ SCO が cmi.core.exit に “suspend” を設定した場合には, LMS は次の SCO 起動時に, cmi.core.entry に “resume” を設定しなければならない. ・ SCO が cmi.core.exit に “logout” を設定した場合には, LMS は次の SCO 起動時に, cmi.core.entry に “” (空文字列) を設定しなければならない. さらに, cmi.core.exit に “logout” を設定した SCO が LMSFinish() を呼び出すかユーザが抜け出す場合, LMS は学習者をコースから退出させなければならない. ・ SCO が cmi.core.exit に “” (空文字列) を設定した場合には, LMS は次の SCO 起動時に, cmi.core.entry に “” (空文字列) を設定しなければならない. ・ SCO が cmi.core.exit に何の値も設定しない場合には, LMS は次の SCO 起動時に, cmi.core.entry に “” (空文字列) を設定しなければならない. ・ LMSGetValue(): LMS は下記に従ってエラーコードを設定し, 空文字列(“”)を返すべきである. <ul style="list-style-type: none"> ・ エラーコード: <ul style="list-style-type: none"> ・ 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. 注意: この要素は (サポート) 必須でないので, LMS はこれをサポートしなければならない. ・ 404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue()を呼ぼうとする場合, LMS はエラーコードを 404 に設定し, 空文字列(“”)を返さなければならない. ・ LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSSetValue(“cmi.core.exit” , “logout”) ・ 設定値の例: "time-out"
--	--

	<p>"suspend" "logout"</p> <p>・エラーコード:</p> <p>405 — 不正データ・タイプ. この要素がサポートされ (この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない) , リクエストが, 正しいデータ型ではない値で LMSSetValue()を起動しようとする場合.</p> <p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. 注意: この要素は (サポート) 必須でなので, LMS はこれをサポートしなければならない.</p> <p>SCO の使用例:</p> <p>LMSSetValue("cmi.core.exit","time-out")</p>
cmi.core.session_time	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMITimespan</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>定義: これは, 学習者が SCO を退出するまでに費やした合計時間を時, 分, 秒で表したものである. すなわち, これはセッションの初めから終わりまでの, SCO の一回の使用時間を表す.</p> <p>使用法: SCO の 1 セッションで費やした時間を記録するために使用される. LMS は cmi.core.total_time を決定する際にこれを使用する.</p> <p>形式: 次の数値形式で示される時, 分および秒での時間の長さ:HHHH:MM::SS.SS</p> <p>時間は最低 2 桁の数字および最高 4 桁の数字を持つ. 分はちょうど 2 桁の数字から成る. 秒は, 2 桁の数字とオプションの小数点として 1 桁または 2 桁の追加の数字を含む. (すなわち, 34.45).</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 要素は LMS によって初期化される必要はない. この要素に対しては, LMSSetValue()呼び出しはない. SCO がこの要素をコントロールする. ・ 追加動作: SCO は一回の実行中に, 複数回 cmi.core.session_time を実行できる. SCO が LMSFinish()を発行するか, 利用者が SCO から離れた場合, LMS は, SCO が設定した (1 組あるとして) 最後の cmi.core.total_time を取得して, その時間を cmi.core.session_time に積算しなければならない. SCO が引き続いて起動され, cmi.core.total_time について LMSSetValue()呼び出された場合, LMS は積算された時間を返さなければならない. LMS は, cmi.core.total_time について LMSSetValue()呼び出しがあった

	<p>場合、LMS に送られた複数時間を積算してはならない。 cmi.core.total_time について LMSSetValue の複数回呼び出しがあった場合、LMS は cmi.core.session_time として保持している値を上書きしなければならない。</p> <ul style="list-style-type: none"> • LMSErrorValue():LMS は下記に従ってエラーコードを設定し、空文字列("")を返すべきである。 <ul style="list-style-type: none"> • エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 <i>注意:</i> この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 404 — 要素は書き込み専用. SCO がこの要素に対して LMSErrorValue()を呼ぼうとする場合、LMS はエラーコードを 404 に設定し、空文字列("")を返すべきである。 • LMSSetValue(): データモデル要素に与えられた値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 • API 呼び出しの例: LMSSetValue("cmi.core.session_time" , "0010:34:34.56") • 設定値の例: "0010:34:34.56" "05:15:00" • エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ. LMSSetValue()が起動され、要素に設定されようとする値が正しいデータ・タイプ(Data Type)でない場合。 401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。 <i>注意:</i> この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。 <p>SCO の使用例:</p> <p>LMSSetValue("cmi.core.session_time","0000:12:30")</p>
--	--

cmi.suspend_data	
<p>Supported API calls:</p> <p>LMSErrorValue() LMSSetValue()</p>	<p>定義: SCO によって前利用時に生成され、現利用時に必要となる独自情報。この独自情報は、SCO 起動時に適用される。この要素は、通常、再開情報として SCO に使用される。この独自情報は SCO によって作成され、次に SCO が実行される時に、SCO に戻さ</p>

<p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMISString4096</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>れるように、LMS によって格納される。</p> <p>LMS は、各学習者の SCO ごとに、このグループのためのスペースをとっておかなければならない。LMS はこのデータを格納し、SCO が再び実行される場合、SCO にこのデータを返す。学習者がコースにいる限り、LMS はこのデータを保持する。</p> <p>使用法: SCO の再開時にのみ有効。学習者が SCO を完了する前に退出すれば、SCO はこの値を設定することができる。その後、SCO は、再開時にこの情報を使用する。</p> <p>形式: SCO 独自。このデータに対するただ唯一の制限は以下のとおりである。</p> <ol style="list-style-type: none"> 1. データは ASCII フォーマットで転送されなければならない。その後、SCO は要求された形式に変換する。 2. LMS の過度の負荷を避けるために、この要素は 4096 バイトのデータ（量）に制限されるべきである。 <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS は、この要素を空白(空文字列)へ初期化しなくてはならない。SCO はこの要素を設定する責任範囲がある。SCO がこの値を設定する前に、LMSGetValue()が要求される場合、LMS は空文字列(“”)を返すべきである。 ● LMSGetValue(): データモデルに格納された値を返す。返り値は正しいデータ・タイプでなければならない。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。注意: この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。 ● LMSSetValue(): データモデル要素に指定された値を設定する。値は、この要素のデータ・タイプに合致しなければならない。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 405 — 不正データ・タイプ。 LMSSetValue()が起動され、要素に設定されようとする値が正しいデータ・タイプ(Data Type)でない場合。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。注意: この要素は（サポート）必須でなので、LMS はこれをサポートしなければならない。 ● 返り値/設定値の例: SCO の実装に依存。 <p>SCO の使用例:</p>
--	--

	<pre>// Upon re-launch of a SCO by a student // Get the Suspend Data var suspendData = LMSGetValue("cmi.suspend_data") if (LMSGetLastError() == 0) { // Use suspend data }</pre>
--	--

cmi.launch_data	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: Yes</p> <p>データ・タイプ: CMISString4096</p> <p>SCO のアクセス可能性: 読み取りのみ</p>	<p>定義: SCO 生成時に生成された独自情報で、毎回の使用のために必要。この情報なしでは走行しえない。</p> <p>使用法: cmi.launch_data は、SCO の起動を支援するために利用される。この値は、指定された SCO に対して常に同じである。</p> <p>形式: テキスト・フィールド。これは、SCO が正しく機能するのに必要ないかなるシステム独自情報をも含む。このフィールドは 4096 文字までに制限する。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: この値は、マニフェストを使用して、LMS によって初期化されるべきである。LMS は(adlcp:datafrimlms) マニフェスト要素を使用すべきである。マニフェスト中に起動データが見つからない場合、起動データは空文字列(“”)に設定されるべきである。 ● LMSGetValue(): データモデルに格納された値を返す。返り値は正しいデータ・タイプでなければならない。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 ー 非実装エラー。 この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 <i>注意: この要素は (サポート) 必須でなので、LMS はこれをサポートしなければならない。</i> ● LMSSetValue(): LMS は下記に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 403 ー 要素は読み出し専用。 この要素に対して、LMSSetValue()を起動する要求があった場合、LMS はエラーコードを 403 に設定すべきである。 401ー 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。

	<p>SCO の使用例:</p> <pre> var launchData = LMSGetValue("cmi.launch_data") if (LMSGetLastError() == 0) { // use launch_data } </pre>
<p>cmi.comments</p> <p>SCO のためのコメントを集めて配布するためのメカニズム.</p> <p>cmi.comments - SCO がコメントを獲得／設定することができることを表す. cmi.comments_from_lms - SCO(SCO の観点からは読み出し専用)にコメントを提供することができることを表す.</p>	
<p>cmi.comments</p>	
<p>Supported API calls:</p> <p>LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISString4096</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: SCO からの自由なフィードバック. 例えば, 学習者は, SCO の中のいかなる時点でもコメントを残すことができる. あるいは, 彼らは SCO の終わりにコメントを求められるかもしれない. コメントは, さらに SCO 内のどこで, いつそれが作成されたかの指標を有する. 位置は, コメントの中にタグ付けされて埋め込まれる.</p> <p>使用法: SCO が SCO に関するコメントを LMS に送るようになるために使用される. 学習者が入力したコメントを収集するために使用することができる.</p> <p>形式: 英数字テキストおよび特殊文字からなる自由形式.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: なし. 空文字列("")に初期化しなくてはならない. SCO はこの値を設定する責務がある. SCO がこの値を設定する前に, LMSGetValue() が呼ばれる場合, LMS は空文字列("")を返す. ● LMSGetValue(): データモデルに格納された値を返す. 戻り値は正しいデータ・タイプでなければならない. もし何のコメントも提供されなければ, LMS は空文字列("")を返すべきである. <ul style="list-style-type: none"> • エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> ● LMSSetValue(): データモデル要素に与えられた値を設定する. LMSSetValue() の要求はコメントを設定すること. (複数)コメントは連結されるべきである.

	<ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. 405 — 不正データ・タイプ. この要素がサポートされ, リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合. ・ 返り値/設定値の例: SCO の実装に依存. SCO の使用例: <pre>var comments = LMSGetValue("cmi.comments") if (LMSGetLastError() == 0) { // use launch_data }</pre>
cmi.comments_from_lms	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISString4096</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: この要素は, LMS からのコメントを表す. これが使用される例は, インストラクター・コメントの形式である. これらのタイプのコメントは, 適切な場合に SCO が学習者に向けて提示する.</p> <p>使用法: LMS からの SCO と関連するいかなるコメントも SCO が参照できるように使用される.</p> <p>形式: 特定の形式はない.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 提供される場合は, これらのコメントは LMS によって初期化される. ・ LMSGetValue(): データモデルに格納された値を返す. 返り値は正しいデータ・タイプでなければならない. もし何のインストラクター・コメントもなければ, LMS は空文字列(“”)を返すべきである. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue(“cmi.comments_from_lms”) ・ 返り値の例: 実装に依存. ・ エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すため

	<p>にエラーコードが設定される.</p> <ul style="list-style-type: none"> ● LMSSetValue(): LMS は以下に従ってエラーコードを設定しなくてはならない. <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 403 — 要素は読み出し専用. この要素に対して, LMSSetValue()を起動する要求があった場合, LMS はエラーコードを 403 に設定すべきである. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. <p>SCO の使用例:</p> <pre>// The SCO may want to see if there are any comments from the // LMS for the student. var commentsFromLMS = LMSGetValue("cmi.comments_from_lms"); if (LMSGetLastError() == "0") { // use the comments from the LMS }</pre>
--	---

cmi.objectives SCO でカバーされた個々の学習目標を学習者がどう実行したかを識別する. cmi.objectives の子: id, score, status	
cmi.objectives._children	
Supported API calls: LMSGetValue() LMS Mandatory: No データ・タイプ: CMISString255 SCO のアクセス可能性: 読み取りのみ	<p>定義: _children キーワードは LMS によってサポートされる cmi.objectives カテゴリの要素をすべて決定するために使用される. サポートされていても, 要素に子要素がない場合, 空文字列が返される. この要素がサポートされない場合, 返り値はない. 次に最後のエラーを要求すれば, 要素がサポートされないことを確認できる.</p> <p>使用法: どの要素が LMS によってサポートされているかを決定するために使用される.</p> <p>形式: 返り値は LMS によってサポートされる cmi.objectives カテゴリ中のすべての要素名をコンマで区切ったリストである. 学習者が複雑なシナリオ (例えばシミュレーション) で学習する場合, どのような詳細情報が集められるかを識別する.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: このグループでサポートされる子のセット.

	<p>LMSGetValue() リクエストの結果，サポートされる子の適切なリストが返される．</p> <ul style="list-style-type: none"> ● LMSGetValue(): サポートされる要素をコンマで区切ったリストを返す． <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.objectives._children.") • 返り値の例: "id,score" "id,status" • エラーコード: 401 — 非実装エラー．この要素がサポートされないならば，空文字列が返され，そしてこの要素がサポートされていないことを示すためにエラーコードが設定される． <p>LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである．</p> <ul style="list-style-type: none"> • エラーコード: 402 — 無効な設定値，要素はキーワードである．要素が LMS によってサポートされ，この要素に対して LMSSetValue() を起動しようとする場合，LMS はエラーコードを 402 に設定すべきである． 401 — 非実装エラー．この要素がサポートされないならば，LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する． <p>SCO の使用例:</p> <p>SCO はどの要素が LMS によってサポートされているかを決定するためにこの要素を使用することができる．</p> <pre>var objChildren = LMSGetValue("cmi.objectives._children") if (objChildren.indexOf("status") != -1) { // Set the Objectives Status }</pre>
cmi.objectives._count	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIIInteger</p>	<p>定義: _count キーワードは cmi.objectives リスト中の現在のレコード数を決定するために使用される．エントリーの総数が返される．SCO が cmi.objectives レコードの数を知らないならば，現在の学習者カウント数を 0 で開始することができる．これにより，第 1 のインデックス位置に現在格納されている学習目標に関する情報が上書きされる．上書きするか追加するかは，作成者が SCO を作成するときに決定される．</p> <p>使用法: LMS によって格納された学習目標の数を決定するために使用される．SCO はどの対象レコードを検索するかを決めるためにこの数</p>

<p>SCO のアクセス可能性:読み出し専用</p>	<p>を使用することができる。「3」（という数字）が返される場合、SCO は、レコードの 0 から 2 までを参照できることが分かる。</p> <p>形式: 現在の要素リストか配列中のアイテムの数を示す整数として値を返す。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS は SCO の最初の起動時に、この値を 0 で初期化しなくてはならない。SCO は学習目標を一つも挿入しない。 ● LMSGetValue(): LMS によって格納された学習目標エンTRIES の総数を返す。 <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue("cmi.objectives._count") ・ 返り値の例: "0" "4" ・ エラーコード: 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> ・ エラーコード: 402 — 無効な設定値、要素はキーワードである。要素が LMS によってサポートされ、この要素に対して LMSSetValue() を起動しようとする場合、LMS はエラーコードを 402 に設定すべきである。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。 <p>SCO の使用例: SCO は、後にプロセスの中でどの配列インデックスを使用するかを決めるために _count 要素を使用する。</p> <pre>// get the count of objectives recorded by the LMS var totalObj = LMSGetValue("cmi.objectives._count") // The value return from the LMS is the total number record // subtract one from the total (to compensate for the zero based array) var request = "cmi.objectives." + totalObj - 1 + ".id" // Call set on the new Objective ID LMSSetValue(request, "Obj1_110")</pre>
	<p>cmi.objectives.n.id</p>

<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIIIdentifier</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: 学習目的に対して、開発者が定義する内部的な SCO 固有の識別子.</p> <p>使用法: 学習目的を識別するための方法.</p> <p>形式: 英数字文字列. 空白は含まず.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 空文字列(“”)に初期化. SCO はこの値を設定しなくてはならない. SCO がこの値を設定する前に, LMSGetValue()が呼ばれる場合, LMS は空文字列(“”)を返さなければならない. ● LMSGetValue(): LMS に格納された学習目標 ID を返す. 返された値は関連するデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue(「cmi.objectives.0.id」) ・ エラーコード: 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ エラーコード: 405 — 不正データ・タイプ(Incorrect Data Type). この要素がサポートされ, リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue()の起動要求をする場合. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. ・ 返り値/設定値の例: “A1317” “Obj123” <p>SCO の使用例: SCO は, 学習目的に関するサマリー情報を表示したいとする.</p> <pre>var objID = LMSGetValue("cmi.objectives.0.id") // Display the objective ID for the SCO</pre> <p>SCO は理由を報告するために学習目的 ID を設定しようとする.</p> <pre>LMSSetValue("cmi.objectives.0.id","Obj1")</pre>
---	---

cmi.objectives.n.score	
各学習目的は関連する点数（スコア）を含むことができる。	
cmi.objectives.n.score の子:	
raw, min, max	
cmi.objecitves.n.score._children	
Supported API calls: LMSGetValue()	定義: _children キーワードは LMS によってサポートされる cmi.objectives.n.score カテゴリの要素すべてを決定するために使用される。サポートされていても要素に子要素がない場合、空文字列が返される。この要素がサポートされない場合、戻り値はない。次に最後のエラーを要求すれば、要素がサポートされないことを確認できる。
LMS Mandatory: No	使用法: どの要素が LMS によってサポートされるかを決定するために使用される。
データ・タイプ: CMISString255	形式: 戻り値は LMS によってサポートされる cmi.objectives.n.score カテゴリのすべての要素名をコンマで区切ったリストである。
SCO のアクセス可能性: 読み出し専用	LMS の動作: <ul style="list-style-type: none"> ・ 初期化: このグループでサポートされる子要素のセット。その結果、LMSGetValue() リクエストでは、サポートされた子要素の適切なリストが返される。 ・ LMSGetValue(): サポートされた要素のコンマで区切られたリストを返す。 ・ API 呼び出しの例: LMSGetValue(「cmi.objectives.0.score._children.」) ・ 戻り値の例: <pre> "raw,min,max" "raw" "" </pre> ・ エラーコード: 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ・ LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> ・ エラーコード: 402 — 無効な設定値、要素はキーワードである。要素が LMS によってサポートされ、この要素に対して LMSSetValue() を起動しようとする場合、LMS はエラーコードを 402 に設定すべきである。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示

	<p>すためにエラーコードを 401 に設定する.</p> <p>SCO の使用例: SCO は, どの要素が LMS によってサポートされているかを決定するためにこの要素を使用することができる.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("raw") != -1) { // Set the Objectives score }</pre>
cmi.objectives.n.score.raw	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIDecimal または CMIBlack</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: 学習目標に対する学習者の各試行後の学習(結果)の数的表現. 加工されていない素点 (そのもの). Cmi.objectives.n.score.raw は, 0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習目標に対する各試行後の素点.</p> <p>形式: 10 進数あるいは空白.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 要素は空文字列("")に初期化されなくてはならない. SCO はこの値を設定しなくてはならない. SCO がこの値を設定する前に, LMSGetValue()の呼び出しがある場合, LMS は空文字列("")を返すべきである. ・ LMSGetValue(): LMS は, リクエストによって識別される学習目標の素点を返すべきである. 戻り値は, 関連するデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue("cmi.objectives.0.score.raw") ・ エラーコード: 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ・ LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSSetValue("cmi.objectives.0.score.raw" , "5") ・ エラーコード: 405 — 不正データ・タイプ. この要素がサポートされ, リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue()の起動要求をする場合. 401 — 非実装エラー. この要素がサポートされないな

	<p>らば、LMSはこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する.</p> <ul style="list-style-type: none"> • 返り値/設定値の例: "96.7" "5" "" <p>SCO の使用例: SCO は学習目標に関連した素点を設定するためにこれを使用することができる.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("raw") != -1) { LMSSetValue("cmi.objectives.0.score.raw","85") }</pre>
cmi.objectives.n.score.max	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIDecimal または CMIBlank</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: 学習者が学習目標に対して取得する可能性があった最高得点または総数.</p> <p>cmi.objective.n.score.max は、0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習目標に対する各試行の後の最高得点.</p> <p>形式: 10 進数あるいは空白.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: 要素は空文字列("")に初期化されなくてはならない. SCO はこの値を設定しなくてはならない. SCO がこの値を設定する前に、LMSGetValue()が呼ばれる場合、LMS は空文字列("")を返さなくてはならない. • LMSGetValue(): LMS は、リクエストによって識別される学習目標の最高得点を返すべきである. 返り値は関連するデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.objectives.0.score.max") • エラーコード: 401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. • LMSSetValue(): データモデル要素に指定された値を設定する. 値は、この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> • API 呼び出しの例:

	<p>LMSSetValue(“cmi.objectives.0.score.max” , “10”)</p> <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 405 — 不正データ・タイプ. この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合. 401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. ・ 返り値／設定値の例: <pre>"100" "10" ""</pre> <p>SCO の使用例: SCO は、学習目標に関連した最高得点を設定するためにこれを使用することができる.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("max") != -1) { LMSSetValue("cmi.objectives.0.score.max","5") }</pre>
cmi.objectives.n.score.min	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIDecimal または CMIBlank</p> <p>SCO のアクセス可能性:リード/書き込み</p>	<p>定義: 学習者が学習目標に対して取得する可能性があった最低得点.</p> <p>cmi.objective.n.score.min は、0 から 100 の間の正規化された値でなければならない.</p> <p>使用法: 学習目標に対する各試行の後の最低点.</p> <p>形式: 10 進数あるいは空白.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: 要素は空文字列(“”)に初期化されなくてはならない. SCO はこの値を設定しなくてはならない. SCO がこの値を設定する前に、LMSGetValue() が呼ばれる場合、LMS は空文字列(“”)を返さなくてはならない. ・ LMSGetValue(): LMS は、リクエストによって識別される学習目標の最低得点を返すべきである. 返り値は関連するデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue(“cmi.objectives.0.score.min” , “5” ・ エラーコード: <ul style="list-style-type: none"> 405 — 不正データ・タイプ. この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合.

	<p>401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する.</p> <p>返り値／設定値の例:</p> <pre>"0" "45.5" ""</pre> <p>SCO の使用例: SCO は、学習目標に関連した最低得点を設定するためにこれを使用することができる.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("min") != -1) { LMSSetValue("cmi.objectives.0.score.min","2") }</pre>
cmi.objectives.n.status	
<p>Supported API calls LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory : No</p> <p>データ・タイプ: CMIVocabulary(status)</p> <p>"passed" "completed" "failed" "incomplete" "browsed" "not attempted"</p> <p>SCO のアクセス可能性: 読み取り/書き込み</p>	<p>定義: SCO の学習目標を達成しようとする各試行後に、学習者が獲得した SCO の学習目標の(習得)状況.</p> <p>可能なボキャブラリ値は次の 6 つ: 合格, 完了, 不合格, 非完了, 未実施, 閲覧済み.</p> <p>使用法: 与えられた学習目標に対しての学習者の(習得)状況を追跡するために使用される.</p> <p>形式: ボキャブラリセット. 6 つの取りうるボキャブラリ値:</p> <ul style="list-style-type: none"> • "passed"(合格): SCO 中の学習目標の必要数が達成された, あるいは, 必要点数が達成された. 学習者は SCO を完了したとみなされ, 合格した. • "completed"(完了): 学習目標は合格か不合格のいずれかだが(分からない), 学習者は学習目標を体験した. 学習者は学習目標を完了したとみなされる. 例えば, 合格は, LMS システムに既知の特定の得点に依存する. SCO は素点を知っているが, その素点が学習目標の合格に十分な高得点かは分からない. • "failed"(不合格): 学習目標は合格とならなかった. 学習目標は学習者によって完了したかどうかはわからない. 学習者は学習目標に関連した(教材)資料を体験したとはみなされるが, 不合格だった. • "incomplete"(非完了): 学習者は学習目標に関連した(教材)資料に着手したが, 完了しなかった. • "browsed"(閲覧済み): 学習者は, 最初の試行で LMS モードの「閲覧」で学習目標と関連する教材を持つ SCO を起動した.

- **“not attempted(未実施):** 学習者が SCO を試行しようとしたが、何かの理由でそれを終わることができなかったこと(非完了)を意味する。「未実施」は学習者が学習目標を含む SCO を開始することすらしなかったことを意味する。多分、学習者はただ目次を読んだか、SCO の概要を読んだだけで、自分の学習準備ができていないと判断した。SCO 内のいくつかのアルゴリズムは、いつ学習目標が「未実施」から「非完了」へ移行するかを SCO が決定するために使用される。

LMS の動作:

- **初期化:** SCO によって処理される。
- **LMSGetValue():** LMS は、リクエスト(処理要求)によって識別される学習目標の状況を返すべきである。返り値は関連するデータの型と合致しなければならない。
 - **API 呼び出しの例:**
LMSGetValue(“cmi.objectives.0.status”)
 - **エラーコード:**
 - 401** — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。
- **LMSSetValue():** 値は、この要素のデータ・タイプと合致しなければならない。
 - **API 呼び出しの例:**
LMSSetValue(“cmi.objectives.0.status”, “failed”)
 - **エラーコード:**
 - 405** — 不正データ・タイプ。この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合。
 - 401** — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。
- **返り値/設定値の例:**
“not attempted”
“passed”
“browsed”

SCO の使用例:

SCO は学習目標に関連したステータス（学習状況）を設定するためにこれを使用する。

```
var objScoreChildren = LMSGetValue("cmi.objectives._children")
if (objScoreChildren.indexOf("status") != -1)
{
  LMSSetValue("cmi.objectives.0.status","failed")
}
```

cmi.student_data	
学習者の学習実行に基づいた SCO のカスタマイズをサポートする情報。	
cmi.student_data の子:	
mastery_score, max_time_allowed, time_limit_action	
cmi.student_data._children	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: _children キーワードは LMS によってサポートされる student_data カテゴリの要素をすべて決定するために使用される。サポートはされていても要素に子要素がない場合、空文字列が返される。この要素がサポートされない場合、空文字列が返される。返り値はない。次に最終のエラーを要求すれば、要素がサポートされないことを確認できる。</p> <p>使用法: どの cmi.student_data データ要素が LMS によってサポートされているかを決定する。</p> <p>形式: 返り値は LMS によってサポートされる student_data カテゴリ要素名のすべてをコンマで区切ったリストである。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: このグループでサポートされる子要素のセット。その結果、LMSGetValue()呼び出し要求では、サポートされた子要素の適切なリストが返される。 ・ LMSGetValue(): LMS は、サポートされた要素をコンマで区切ったリストを返す。 <ul style="list-style-type: none"> ・ 返り値の例: <pre>"mastery_score,time_limit_action,max_time_allowed" "max_time_allowed"</pre> ・ エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ・ LMSSetValue(): LMS は以下に従ってエラーコードを設定しなくてはならない。 <ul style="list-style-type: none"> ・ エラーコード: <ul style="list-style-type: none"> 402 — 無効な設定値、要素はキーワードである。要素が LMS によってサポートされ、この要素に対して LMSSetValue()を起動しようとする場合、LMS はエラ

	<p>ーコードを 402 に設定すべきである。</p> <p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する.</p> <p>SCO の使用例:</p> <pre> var studentDataChildren = LMSGetValue("cmi.student_data.children"); if (studentDataChildren.indexOf("attempt_number") != -1) { var maxTimeAllowed = LMSGetValue("cmi.student_data.max_time_allowed"); } </pre>
cmi.student_data.mastery_score	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIDecimal</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: SCO の外部で決定される合格に必要な得点. SCO スコアが合格点以上である場合, 学習者は合格したとみなされるか, あるいは内容を習得したとみなす. 合格点が LMS システムによって決定されるため, 合格に必要な得点がいくつであるか SCO が知らない場合もある.</p> <p>使用法: mastery_score を LMS システムがサポートするためには, SCO から渡された得点に基づいて lesson_status を変更することができなければならない. 単に mastery_score を SCO へ渡すだけでは, この機能を全面的にサポートしたことにはならない.</p> <p>形式: 10 進数</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ・ 初期化: LMS はマニフェスト(adlcp: masteryScore)の項目要素から得られた値に設定しなくてはならない. ・ LMSGetValue(): LMS によって格納された現在値を返す. <ul style="list-style-type: none"> ・ 返り値の例: "75" "100" "5" ・ エラーコード: 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ・ LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである. <ul style="list-style-type: none"> ・ エラーコード:

	<p>403 — 要素は読み出し専用. この要素を LMS がサポートしており, この要素に対して, LMSSetValue() を起動する要求があった場合, LMS はエラーコードを 403 に設定すべきである.</p> <p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する.</p> <p>SCO の使用例:</p> <pre>var masteryScoreValue = LMSGetValue("cmi.student_data.mastery_score");</pre>
cmi.student_data.max_time_allowed	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMITimespan</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: 現在の SCO における試行で学習者が利用できる総時間. 時間が超過した場合の SCO の応答に関しては, time_limit_action を参照されたい.</p> <p>使用法: 学習者が SCO 中にいることを認められた最大時間を SCO に提示するために使用される.</p> <p>形式: コロンによって区切られた時, 分および秒. HHHH:MM:SS.SS</p> <p>「時」は最低 2 桁, 最高 4 桁の数字を持つ. 「分」はちょうど 2 桁の数字から成なる. 「秒」は 2 桁の数字と, オプションで小数点と 1 桁あるいは 2 桁の追加の数字(つまり 34.45 のような)を含む.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS はマニフェスト(adlcp: masteryScore)の項目要素から得られた値に設定しなくてはならない. ● LMSGetValue(): LMS によって格納された現在値を返す. <ul style="list-style-type: none"> ・ 返り値の例: <pre>"00:14:30" "02:03:00" "01:09:00"</pre> ・ エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである. <ul style="list-style-type: none"> ・ エラーコード: <p>403 — 要素は読み出し専用. この要素を LMS がサポートしており, この要素に対して, LMSSetValue() を</p>

	<p>起動する要求があった場合、LMS はエラーコードを 403 に設定すべきである。</p> <p>401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。</p> <p>SCO の使用例:</p> <pre>var maxTimeAllowedValue = LMSGetValue("cmi.student_data.max_time_allowed");</pre>
cmi.student_data.time_limit_action	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIVocabulary(タイム・リミット・アクション)</p> <p>"exit,message" "exit,no message" "continue,message" "continue,no message"</p> <p>SCO のアクセス可能性: 読み出し専用です</p>	<p>定義: 最大許容時間(max_time_allowed)を超過する場合、どうすべきかを SCO に伝える。この要素に対しては次の 2 つの引数がある。</p> <p>SCO がすべきこと — 退出あるいは継続 学習者が見るべきもの — メッセージありあるいはメッセージなし</p> <p>使用法: 最大許容時間を超過した場合、SCO がすべきアクションを指示するために利用される。</p> <p>形式: ボキャブラリセット。4 つの可能なボキャブラリ値は以下。</p> <ul style="list-style-type: none"> ● "exit,message" ● "exit,no message" ● "continue,message" ● "continue,no message" <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS は マニフェスト(adlcp:timelimitaction)項目要素から得られた値に設定しなくてはならない。 ● LMSGetValue(): LMS によって格納された現在値を返す。 <ul style="list-style-type: none"> ・ 返り値の例: <pre>"exit,message" "continue,no message"</pre> ・ エラーコード: <p>401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。</p> ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> ・ エラーコード: <p>403 — 要素は読み出し専用。この要素を LMS がサポートしており、この要素に対して、LMSSetValue() を起動する要求があった場合、LMS はエラーコードを</p>

	<p>403 に設定すべきである。</p> <p>401 – 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。</p> <p>SCO の使用例:</p> <pre>var timeLimitActionValue = LMSGetValue("cmi.student_data.time_limit_action");</pre>
--	---

cmi.student_preference 後続の SCO への適切な選択オプション cmi.student_preference の子: audio,language,speed,text	
cmi.student_preference._children	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: _children キーワードは LMS によってサポートされる student_preference カテゴリの要素をすべて決定するために使用される。サポートはされていても、要素に子要素がない場合、空文字列が返される。この要素がサポートされない場合、返り値はない。次に最終のエラー後を要求すれば、要素がサポートされないことを確認できる。</p> <p>使用法: どの cmi.student_preference データ要素が LMS によってサポートされるかを決定する。</p> <p>形式: 返り値は LMS によって支援されるすべての student_preference カテゴリ要素名をコンマで区切ったリストである。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: このグループでサポートされる子要素のセット。その結果、LMSGetValue() 要求では、サポートされた子要素の適切なリストが返される。 ● LMSGetValue(): LMS は、サポートされる要素をコンマで区切ったリストで返す。 <ul style="list-style-type: none"> ・ 返り値の例: "audio, language, speed" "language" ・ エラーコード: <p>401 – 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。</p> ● LMSSetValue(): LMS は以下に従ってエラーコードを設定

	<p>すべきである。</p> <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 402 — 無効な設定値, 要素はキーワードである. 要素が LMS によってサポートされ, この要素に対して LMSSetValue() を起動しようとする場合, LMS はエラーコードを 402 に設定すべきである. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. <p>SCO の使用例:</p> <pre>var studentPreferenceChildren = LMSGetValue("cmi.student_preference._children"); if (studentPreferenceChildren.indexOf("audio") != -1) { LMSSetValue("cmi.student_preference.audio", "10"); }</pre>
cmi.student_preference.audio	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIIInteger</p> <p>SCO のアクセス可能性: 読み出し 書き込み</p>	<p>定義: 学習者は, 音声機能の電源を切断したり音量を制御したりする. この要素は音声機能の電源の状態を示す.</p> <p>使用法: SCO が学習者の音声機能の好みの状態を設定し, かつ LMS から獲得するために使用される.</p> <p>形式: -1 から 100 までの数字</p> <ul style="list-style-type: none"> -1: オフ (いかなる負の数字でもオフとなる). 0: 無変更ステータス(SCO はそのデフォルトを使用するか, または最後(直前)の SCO でのオーディオのステータスと同じまま). 1 - 100: ボリューム・レベル(1 は弱い, 100 は可能な最大ボリューム). <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: (この要素が)サポートされる場合, LMS はこの値を "0" に初期化すべきである. この値を設定するのは SCO の責任範囲である. SCO がこの値を設定する前に, LMSGetValue() 呼び出しが要求される場合, LMS は空文字列("")を返すべきである. ● LMSGetValue(): LMS によって格納された現在値を返す. <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ● LMSSetValue(): LMS データ・アイテムにパラメータとし

	<p>て渡された値を設定する。</p> <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 405 — 不正データ・タイプ. この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合. 401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. ・返り値／設定値の例: <ul style="list-style-type: none"> "-1" "0" "50" <p>SCO の使用例:</p> <pre>LMSSetValue("cmi.student_preference.audio", "10"); var audioValue = LMSGetValue("cmi.student_preference.audio");</pre>
cmi.student_preference.language	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読出し 書き込み</p>	<p>定義: 多言語機能を備えた SCO では、どの言語で情報を伝えるかを指定するためにこの要素を使用すべきである。</p> <p>使用法: SCO が学習者の言語選択を設定し、かつ LMS から獲得するために使用される。</p> <p>形式: 英字列、空白を含んでよい。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: (この要素が)サポートされる場合、LMS は、この値を空文字列("")に初期化すべきである。この値を設定するのは SCO の責任範囲である。SCO がこの値を設定する前に、LMSGetValue() 呼び出しが要求される場合、LMS は空文字列("")を返すべきである。 ● LMSGetValue(): LMS によって格納された現在値を返す。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ● LMSSetValue(): パラメータとして渡された値を LMS データ項目に設定する。 <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 405 — 不正データ・タイプ. この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合.

	<p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する</p> <p>・ 返り値/設定値の例: "English" "French"</p> <p>SCO の使用例:</p> <pre>LMSSetValue("cmi.student_preference.language", "English"); var languageValue = LMSGetValue("cmi.student_preference.language");</pre>
cmi.student_preference.speed	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISInteger</p> <p>SCO のアクセス可能性: 読出し 書き込み</p>	<p>定義: SCO は, そのペースによっては, 理解するのが難しい場合がある. この要素は, 内容伝達のペースを制御する.</p> <p>使用法: SCO が学習者の速度選択を設定し, かつ LMS から獲得するために使用される.</p> <p>形式: -100 から 100 までの数字 -100 はシステムで可能な最も遅いペース. 0 は無変更状態(SCO はデフォルトを使用. SCO 普通の速度で動作). 100 はシステムで可能な最も早いペース.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS によってサポートされる場合, この要素は "0" に初期化されるべきである. この値を設定するは SCO の責任範囲である. SCO がこの値を設定する前に, LMSGetValue() 呼び出しが要請される場合, LMS は空文字列("")を返すべきである. ● LMSGetValue(): LMS によって格納された現在値を返す. ・ エラーコード: 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ● LMSSetValue(): パラメータとして渡された値を LMS データ項目に設定する. ・ エラーコード: 405 — 不正データ・タイプ. この要素がサポートされ, リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合. 401 — 非実装エラー. この要素がサポートされない

	<p>ならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。</p> <ul style="list-style-type: none"> ● 返り値／設定値の例: “-100” “0” “3” <p>SCO の使用例:</p> <pre>LMSSetValue("cmi.student_preference.speed", "-100"); var speedValue = LMSGetValue("cmi.student_preference.speed");</pre>
cmi.student_preference.text	
<p>Supported API calls: LMSGetValue() LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISInteger</p> <p>SCO のアクセス可能性: 読出し 書き込み</p>	<p>定義: オーディオ用にデザインされた SCO では、オーディオのスイッチを切って、テキスト・ウインドウでオーディオの内容を見ることは可能である。または、オーディオ(のスイッチ)を入れたまま、テキストがオーディオと同時に提示されるようにすることも可能である。または、オーディオとスクリーン・グラフィックスだけが利用できるように、テキスト(表示)を消すことも可能である。この要素は、オーディオテキストが SCO に現れるかどうか確認するものである。</p> <p>使用法: 学習者の LMS テキスト選択を設定し、かつ獲得するために SCO によって使用される。</p> <p>形式: 3つの数字のうちの1つ。 -1: テキストはオフ、表示なし。 0: 状態の変化なし。デフォルト使用。 1: テキストはスクリーン上、学習者に示される。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS によってサポートされる場合、この要素は“0”に初期化されるべきである。この値を設定するのは SCO の責任範囲である。SCO がこの値を設定する前に、LMSGetValue()呼び出しが要請される場合、LMS は空文字列(“”)を返すべきである。 ● LMSGetValue(): LMS によって格納された現在値を返す。 ・エラーコード: 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ● LMSSetValue(): パラメータとして渡された値を LMS データ項目に設定する。 ・エラーコード:

	<p>405 — 不正データ・タイプ. この要素がサポートされ、リクエストが正しいデータ・タイプ値をもたない値で LMSSetValue() の起動要求をする場合.</p> <p>401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する</p> <ul style="list-style-type: none"> ● 返り値／設定値の例: "0" "-1" "1" <p>SCO の使用例:</p> <pre>LMSSetValue("cmi.student_preference.text", "-1"); var textValue = LMSGetValue("cmi.student_preference.text");</pre>
<p>cmi.interactions</p> <p>この文脈では、相互作用は、認識され記録可能な、学習者からコンピュータへの入力あるいは入力の組である。このグループ中の項目は、すべて認識され、記録可能な学習者からの入力と関連している。このカテゴリは、学習者が SCO を習得するときに測定された、各相互作用についての詳細な情報を収集する。</p> <p>cmi.interactions の子:</p> <p>id, objectives, time, type, correct_response, weighting, student_response, result, latency</p>	
<p>cmi.interactions._children</p>	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMISString255</p> <p>SCO のアクセス可能性: 読み出し専用</p>	<p>定義: _children キーワードは LMS によってサポートされる cmi.interactions カテゴリの要素をすべて決定するために使用される。サポートはされていても、要素に子要素がない場合、空文字列が返される。この要素がサポートされない場合、返り値はない。次に最後のエラーを要求すれば、要素がサポートされないことを確認できる。</p> <p>使用法: どの要素が LMS によってサポートされるかを決定する。</p> <p>形式: 返り値は LMS によって支援される cmi.interactions カテゴリ内すべての要素名をコンマで区切ったリストである。学習者が SCO を習得するときに測定される各相互作用に関して、いかなる詳細な情報を集めることができるかを識別する。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: このグループでサポートされる子要素のセット。その結果、LMSGetValue() 要求では、サポートされた子要素の適切なリストが返される。 ● LMSGetValue(): サポートされた要素をコンマで区切ったリストを返す。

	<ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.interactions._children") • 返り値の例: "id,time,type" "id,time,type,correct_responses,student_response" • エラーコード: 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである. <ul style="list-style-type: none"> • エラーコード: 402 — 無効な設定値, 要素はキーワードである. 要素が LMS によってサポートされ, この要素に対して LMSSetValue() を起動しようとする場合, LMS はエラーコードを 402 に設定すべきである. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する. <p>SCO の使用例:</p> <p>SCO はどの要素が LMS によってサポートされるかを決定するためにこの要素を使用する.</p> <pre>var intChildren = LMSGetValue("cmi.interactions._children") if (intChildren.indexOf("id") != -1) { // Set the Interaction id }</pre>
cmi.interactions._count	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMIIInteger</p> <p>SCO のアクセス可能性: 読み出し専用です</p>	<p>定義: _count キーワードは cmi.interactions リスト中のレコードの現在数を決定するために使用される. エントリーの総数が返される.</p> <p>SCO が cmi.interactions レコードの数を知らない場合, 現在の学習者数を 0 から開始する. これは, 最初のインデックス位置に現在格納されたインタラクション(送信)に関する任意の情報を上書きすることである. 上書きか追加かは SCO を作成するときに, SCO 作成者によって決定される.</p> <p>使用法: LMS によって格納されたインタラクションの数を決定するために使用される. SCO はどのインタラクション記録を設定すべきかを決定するためにこの数を使用する. 「3」がそのとき返された場合, SCO は, レコード 0 から 2 が既に使われており, 次の利用可能なインデックスは「3」だと分かる.</p>

	<p>として返す.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS は, SCO の初期起動時にこれを 0 に初期化しなくてはならない. インタクションデータは SCO によって挿入されていない. ● LMSGetValue(): LMS によって格納された interaction エンタリーの総数を返す. <ul style="list-style-type: none"> ・ API 呼び出しの例: LMSGetValue("cmi.interactions._count") ・ 返り値の例: "0" "4" ・ エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである. <ul style="list-style-type: none"> ・ エラーコード: <p>402 — 無効な設定値, 要素はキーワードである. 要素が LMS によってサポートされ, この要素に対して LMSSetValue() を起動しようとする場合, LMS はエラーコードを 402 に設定すべきである.</p> <p>401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する.</p> <p>SCO の使用例:</p> <p>SCO はプロセスの中でどの配列インデックスをその後使用すべきかを決定するために _count 要素を使用する.</p> <pre>// get the count of interactions recorded by the LMS var totalInteractions = LMSGetValue("cmi.interactions._count") // The value return from the LMS is the total number records var request = "cmi.interactions." + totalInteractions + ".id" // Set the Interaction ID LMSSetValue(request, "Int_110")</pre>
--	---

cmi.interactions.n.id	
<p>Supported API calls:</p> <p>LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ :CMIIIdentifier</p> <p>SCO のアクセス可能性 :書き込み専用</p>	<p>定義: インタラクションの一意的識別子.</p> <p>使用法: 一意なインタラクション id を設定するために使用する. SCO 固有である.</p> <p>形式: 英数字文字列. 内部スペースは含まず.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる. ● LMSGetValue(): LMS は以下に従ってエラーコードを設定すべきであり, 空文字列を返さなければならない. <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. 404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合, LMS はエラーコードに 404 を設定し, 空文字列("")を返すべきである. ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・API 呼び出しの例: <pre>LMSSetValue("cmi.interactions.0.id" , "I_001")</pre> ・設定値の例 : <pre>"I_001" "i1"</pre> ・エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ : LMSSetValue() が起動され, 要素に対して設定される値が, 不正データ・タイプである. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを(401)に設定する. <p>SCO の使用例:</p> <pre>// Set the first interactions id LMSSetValue("cmi.interactions.0.id","I_001");</pre>
cmi.interactions.n.objectives	

cmi.interactions.n.objectives._count

Supported API calls:

LMSGetValue()

LMS Mandatory:

データ・タイプ:
CMIIInteger

SCO のアクセス可能性:
読み出し専用です

定義: `_count` キーワードは `cmi.interactions` 学習目標 `id` リスト中の現在のレコード数を決定するために使用される。エントリーの総数が返される。SCO が `cmi.interactions.n.objectives` レコードの数を知らない場合、現在の学習者数を **0** から始める。これは、最初のインデックス位置に現在格納されている学習目標 `id` に関する情報を上書きすることになる。上書きか追加かは SCO を作成する時に、SCO 作成者によって決定される。

使用法: LMS があるインタラクションに対して格納した学習目標 `id` の数を決定するために使用される。SCO はこの数を、どの学習目標 `id` レコードを設定するか決定に使用する。「3」が返される場合、SCO は、すでに **0** から **2** は占有されているので、次の利用可能なインデックスは「3」だと分かる。

形式: 現在の要素リストないし配列中のアイテム数を示す値を整数として返す。

LMS の動作:

- **初期化:** LMS は、SCO の初期起動時にこの数を **0** に初期化しなくてはならない、学習目標 `id` データは SCO によって挿入されていない。
- **LMSGetValue():** LMS が、あるインタラクションに対して格納した、学習目標 `id` の総数を返す。
 - API 呼び出しの例:
LMSGetValue("cmi.interactions.3.objectives._count")
 - 返り値の例:
"0"
"4"
 - エラーコード:
401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。

LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである。

- エラーコード:
402 — 無効な設定値、要素はキーワードである。要素が LMS によってサポートされ、この要素に対して LMSSetValue() を起動しようとする場合、LMS はエラーコードを **402** に設定すべきである。
401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを **401** に設定する。

	<p>SCO の使用例:</p> <p>SCO はプロセスの中でどの配列インデックスをその後使用するかを決定するために_count 要素を使用する.</p> <pre>// get the count of objective ids recorded by the LMS for a given interaction var totalIDs = LMSGetValue("cmi.interactions.0.objectives_count") // The value return from the LMS is the total number records var request = "cmi.interactions.0.objectives." + totalID + ".id" // Set the Interactions Objective ID LMSSetValue(request, "ObjID-110")</pre>
cmi.interactions.n.objectives.n.id	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMIIIdentifier</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>定義: 開発者が作成した学習目標の識別子.</p> <p>使用法: インタラクションが目的とする学習目標を識別するために使用される.</p> <p>形式: 英数字文字列. 内部スペースは含まず.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる. ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し, 空文字列("")を返す. <ul style="list-style-type: none"> • エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. 404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合, LMS はエラーコードに 404 を設定し, 空文字列("")を返すべきである. ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: LMSSetValue("cmi.interactions.0.objectives.0.id", "A1333") • 返り値の例: "A1333" "Obj123" • エラーコード:

	<p>205 — 不正データ・タイプ: LMSSetValue()が起動され、要素に対して設定される値が、不正データ・タイプである.</p> <p>401 — 非実装エラー. この要素がサポートされないならば、LMSはこの要素がサポートされていないことを示すためにエラーコードを(401に)設定する.</p> <p>SCO の使用例:</p> <pre>// set the objective ID for the given interaction LMSSetValue("cmi.interactions.0.objectives.0.id","A1333")</pre>
cmi.interactions.n.time	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory: No</p> <p>データ・タイプ: CMITime</p> <p>SCO のアクセス可能性:書き込み専用</p>	<p>定義: 学習者のインタラクションがいつ完了したかの表示.</p> <p>使用法: インタラクションのタイムスタンプとして使用.</p> <p>形式: 24 時間表示での時間表示. HH:MM:SS.SS の形式で、時間、分および秒で識別.</p> <p>時間と分はちょうど 2 桁の数字で、秒は 2 桁の数字と、オプションで小数点と 1 桁あるいは 2 桁の追加の数字(つまり 34.45 のような)を含む.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる. ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し、空文字列(“”)を返すべきである. <ul style="list-style-type: none"> ・エラーコード: <p>401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> <p>404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue()を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列(“”)を返すべきである.</p> ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は、この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・API 呼び出しの例: LMSSetValue("cmi.interactions.0.time","12:33:35.5") ・戻り値の例: "12:33:35.5" "22:30:40" ・エラーコード:

	<p>205 — 不正データ・タイプ: LMSSetValue()が起動され、要素に対して設定される値が、不正データ・タイプである。</p> <p>401 — 非実装エラー. この要素がサポートされないならば、LMSはこの要素がサポートされていないことを示すためにエラーコードを(401)に設定する。</p> <p>SCO の使用例:</p> <pre>// Set the Interactions time LMSSetValue("cmi.interactions.0.time","12:33:35.5")</pre>
cmi.interactions.n.type	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>Data Type: CMIVocabulary (Interaction) "true-false" "choice" "fill-in" "matching" "performance" "sequencing" "likert" "numeric"</p> <p>SCO アクセス可能性: 書き込み専用</p>	<p>定義: インタラクションのどのカテゴリが記録されるかの表示. インタラクションのタイプは、インタラクションレスポンスがどう解釈されるかを決定する. 7つの可能な問題タイプが以下に定義されている. タイプはこれらだけに制限されてはいない. 他のタイプの問題もある. しかしながら、これら7つのタイプのうち1つが使用される場合には、これらのタイプと一致する識別子である.</p> <p>使用法: 現在起きているインタラクションのタイプを示す.</p> <p>形式: セットボキャブラリ. 8つの可能なボキャブラリ値.</p> <ul style="list-style-type: none"> • "true-false"(○, ×): 2つの可能な回答による問題. • "choice"(多肢選択): 学習者が選択可能な事前定義済み回答の数が限られた問題. 各回答は、数字か文字である. 1つまたはそれ以上の回答が正しい可能性がある. • "fill-in"(穴埋め): 単純に1語または数語からなる答えをもつ問題. 答え/応答はあらかじめ定められておらず、学習者が(選択式と異なり)作成しなければならない. • "matching"(組合せ): 1組あるいは2組の項目の組を有する問題. これらの組の2つ以上の項目が関連を持つ. 問題に回答するには、関連する項目を見つけて対応付けしなくてはならない. • "performance"(実行): パフォーマンス問題は、ある意味で多項式選択の問題に似ている. しかし、記載された答えを選択する代わりに、学習者はタスクかアクションを実行しなければならない. • "sequencing"(並べ替え): 並べ替え式の問題では、学習者は項目の論理的な順序付けをしなければならない. • "likert"(アンケート): アンケート式の問題は、学習者に連続値の選択肢を提示する. 回答は、一般に学習者の見解もしくは姿勢に基づく. • "numeric"(数値): 小数点あり、または、なしの単純な数値が回答として要求される. 正解は、ある範囲内の単一な数値である. <p>LMS の動作:</p>

	<ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる。 ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し、空文字列("")を返さねばならない。 <ul style="list-style-type: none"> ・エラーコード <ul style="list-style-type: none"> 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 404 — 要素は書き込み専用。SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列("")を返すべきである。 ● LMSSetValue(): データモデル要素に指定された値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 <ul style="list-style-type: none"> ・API 呼び出しの例: <pre>LMSSetValue("cmi.interactions.0.type","likert")</pre> ・設定値の例: <pre>"likert" "true-false" "performance"</pre> ・エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ: LMSSetValue() が起動され、要素に対して設定される値が、不正データ・タイプである。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを(401 に)設定する。 <p>SCO の使用例:</p> <pre>// set the interactions type LMSSetValue("cmi.interactions.0.type","choice");</pre>
cmi.interactions.n.correct_responses	
cmi.interactions.n.correct_responses._count	
<p>Supported API calls: LMSGetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMIIInteger</p> <p>SCO のアクセス可能性:</p>	<p>定義: _count キーワードは、cmi.interactions.n.correct_responses リスト中の現在のレコード数を決定するために使用される。エントリの総数が返される。SCO が cmi.interactions.n.correct_responses レコードの数を知らないならば、現在の学習者数を 0 から開始することができる。これにより、最初のインデックス位置に現在格納されている情報は上書きされる。上書きするか追加するかは、作成者が SCO を作成するときに決定する。</p> <p>使用法: あるインタラクションに関して LMS が格納した correct_responses の数を決定するために使用される。SCO はどの</p>

読み出し専用	<p>correct_responses パターンレコードを設定するかを決定するために、この数を使用する。 “3” が返される場合、SCO は、0 から 2 までのレコードが既に占められているので、次の利用可能なインデックスは “3” であると分かる。</p> <p>形式: 現在の要素リストか配列中のアイテムの数を示す整数値を返す。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: LMS は、SCO の初期起動時にこの数を 0 で初期化しなくてはならない。 correct_responses パターン・データは SCO によって挿入されていない。 ● LMSGetValue(): あるインタラクションに関して LMS によって格納された、correct_responses パターンエントリーの総数を返す。 <ul style="list-style-type: none"> • API 呼び出しの例: LMSGetValue("cmi.interactions.0.correct_responses._count") • 返り値の例: “0” “4” • エラーコード: 401 — 非実装エラー。 この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 ● LMSSetValue(): LMS は以下に従ってエラーコードを設定すべきである。 <ul style="list-style-type: none"> • エラーコード: 402 — 無効な設定値、要素はキーワードである。 要素が LMS によってサポートされ、この要素に対して LMSSetValue() を起動しようとする場合、LMS はエラーコードを 402 に設定すべきである。 401 — 非実装エラー。 この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを 401 に設定する。 <p>SCO の使用例:</p> <p>SCO はプロセスの中でどの配列インデックスを後に使用するべきかを決定するために _count 要素を使用する。</p> <pre>// get the count of objective ids recorded by the LMS for a given interaction var totalPattern = LMSGetValue("cmi.interactions.0.correct_responses._count") // The value return from the LMS is the total number records var request = "cmi.interactions.0.correct_responses." + totalPattern + ".pattern"</pre>
--------	---

	// Set the Interactions correct_responses pattern LMSSetValue(request, "t")
cmi.interactions.n.correct_responses.n.pattern	
Supported API calls: LMSSetValue() LMS Mandatory: No データ・タイプ: CMIFeedback SCO のアクセス可能性: 書き込み専用	<p>定義: インタラクションに対して学習者が行えるレスポンスの記述。正しい回答は 1 つ以上かもしれないし、いくつかの回答は他のものよりもっと正確かもしれない。</p> <p>使用法: これは SCO によって提供されるインタラクションに対する正しい回答である。</p> <p>形式: cmi.interactions.n.type が以下の場合、</p> <p>True-False(○, ×): この場合、パターンは 1 文字か数字である。正しい文字は 0, 1, t および f である。0 は偽に対応する。レスポンスが完全な単語(つまり, “true”)である場合、先頭の文字(t)だけ意味がある。</p> <p>Choise(選択): この場合、パターンはコンマで区切られた 1 つ以上の文字である。整数(0-9), 文字(a-z-)あるいは両方を使用してよい。個々の可能なレスポンスは単一の文字に限られる。26 を越える可能性がある場合、パフォーマンス型のレスポンスが使用されなければならない。</p> <p>fill-in (穴埋め): この場合、パターンは英数字文字列である。先頭の印刷可能な文字の後の空白は意味を持つ。</p> <p>Numeric(数値): この場合、パターンは単一の数である。数は小数点を持つ場合と持たない場合がある。</p> <p>Likert(アンケート): likert 問題に対しては正しくない解答というのではない。フィールドはブランクだけの場合もある。</p> <p>Matching(組合せ): この場合、パターンはピリオドで区切られた識別子のペアである。それぞれの組合せの候補は、ソースとターゲットから成る。</p> <p>Performance(実行): この場合、パターンは 255 文字までが(長さ)制限の英数字のフィールドである。</p> <p>Sequencing(並べ替え): この場合、パターンは任意の並びの要素である。要素の最終位置は正しさを決定するために使用され、どのような順序でそれらを並べたかは関係無い。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる。 ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し、

	<p>空文字列("")を返さねばならない。</p> <ul style="list-style-type: none"> ・エラーコード： <ul style="list-style-type: none"> 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 404 — 要素は書き込み専用。SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列("")を返すべきである。 ● LMSSetValue(): データモデル要素に指定された値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 <ul style="list-style-type: none"> ・API 呼び出しの例: <ul style="list-style-type: none"> cmi.interactions.n.type が一致している場合、 LMSSetValue("cmi.interactions.0.correct_responses.0.pattern","1.c.2.b,3.a,4.d") (のように呼び出す。) 設定値の例: <ul style="list-style-type: none"> "1.c.2.b,3.a,4.d" "t" エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ: LMSSetValue() が起動され、要素に対して設定される値が、不正データ・タイプである。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを(401 に)設定する。 <p>SCO の使用例:</p> <pre>// set the response pattern // Question type is a true-false question LMSSetValue("cmi.interactions.0.correct_responses.0.pattern","t");</pre>
cmi.interactions.n.weighting	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMIDecimal</p> <p>SCO のアクセス可能性:書き込み専用</p>	<p>定義: インタラクシヨンの重要度は異なる。重み付けは、あるインタラクシオンを他と比較した相対的な重要度を指定するために使用される要素である。例えば、第 1 のインタラクシオンが 15 の重みを持っており、第 2 のインタラクシオンが 25 の重みを持っている場合、第 2 のインタラクシオンは、重みを反映したいかなる組み合わせの得点にも、より大きな影響を与える。</p> <p>インタラクシヨンの重要度が等しい場合、各インタラクシオンは同じ重みを持つ。</p> <p>重み 0 は、当該のインタラクシオンを、重み付けされた最終スコアの中で数えてはならないことを示す。</p>

	<p>中で数えてはならないことを示す.</p> <p>使用法: インタラクションは重みを持ってよいし, また, 同様に, 複雑なインタラクション内部の個々のアクション, あるいはレスポンスも重みを持ってよい.</p> <p>形式: 単一の浮動小数点数. 小数点はオプションで, あらゆる重み付けに現われる必要はない.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる. ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し, 空文字列("")を返すべきである. <ul style="list-style-type: none"> ・エラーコード: <ul style="list-style-type: none"> 401 — 非実装エラー. この要素がサポートされないならば, 空文字列が返され, そしてこの要素がサポートされていないことを示すためにエラーコードが設定される. 404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合, LMS はエラーコードに 404 を設定し, 空文字列("")を返すべきである. ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は, この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> ・API 呼び出しの例: <pre>LMSSetValue("cmi.interactions.0.weighting","0.66")</pre> ・設定値の例: <pre>"0.66" "0"</pre> ・エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ: LMSSetValue() が起動され, 要素に対して設定される値が, 不正データ・タイプである. 401 — 非実装エラー. この要素がサポートされないならば, LMS はこの要素がサポートされていないことを示すためにエラーコードを(401 に)設定する. <p>SCO の使用例:</p> <pre>// Set the weighting for the interaction LMSSetValue("cmi.interactions.0.weighting","0");</pre>
cmi.interactions.n.student_response	
Supported API calls LMSSetValue()	定義: インタラクションに対して行える解答 (レスポンス) の記述である. 1 つ以上の正しいレスポンスがあるかもしれないし, また, い

<p>LMS Mandatory:No</p> <p>データ・タイプ: CMIFeedback</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>くつかのレスポンスは他のものより正しいかもしれない。</p> <p>使用法: これはインタラクションに対する実際の学習者レスポンスである。この値は次に、cmi.interactions.n.correct_responses.n.patternと比較される。</p> <p>形式: cmi.interactions.n.type が以下の場合、</p> <p>True-False(○, ×): この場合、1 文字か数字である。正しい文字は 0, 1, t および f である。0 は偽に対応する。レスポンスが完全な単語(つまり, “true”)である場合、先頭の文字(t)だけ意味がある。</p> <p>Choise(選択): この場合、コンマで区切られた 1 つ以上の文字である。整数(0-9)、文字(a-z)あるいは両方を使用してよい。個々の可能なレスポンスは単一の文字に限られる。26 を越える可能性がある場合、パフォーマンス型のレスポンスが使用されなければならない。</p> <p>fill-in (穴埋め) : この場合、英数字文字列である。先頭の印刷可能な文字の後の空白は意味を持つ。</p> <p>Numeric(数値): この場合、単一の数である。数は小数点を持つ場合と持たない場合がある。</p> <p>Likert(アンケート): likert 問題に対しては正しくない解答というのではない。フィールドはブランクだけの場合もある。</p> <p>Matching(組合せ): この場合、パターンはピリオドで区切られた識別子のペアである。それぞれの組合せの候補は、ソースとターゲットから成る。</p> <p>Performance(実行): この場合、255 文字までが(長さ)制限の英数字のフィールドである。</p> <p>Sequencing(並べ替え): この場合、パターンは任意の並びの要素である。要素の最終位置は正しさを決定するために使用され、どのような順序でそれらを並べたかは関係無い。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる。 ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し、空文字列(“ ”)を返さねばならない。 <ul style="list-style-type: none"> ・エラーコード: 401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される。 404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列(“ ”)を返すべき
--	---

	<p>である。</p> <ul style="list-style-type: none"> ● LMSSetValue(): データモデル要素に指定された値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 <ul style="list-style-type: none"> ・ API 呼び出しの例: <code>cmi.interactions.n.type</code> が一致している場合、 LMSSetValue("cmi.interactions.0.correct_responses.0.pattern","1.c,2.b,3.a,4.d") (のように呼び出す。) ・ 設定値の例: "t" "1.c,2.b,3.a,4.d" ・ エラーコード: 205 — 不正データ・タイプ: LMSSetValue() が起動され、要素に対して設定される値が、不正データ・タイプである。 <p>c c c c c 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを(401)に設定する。</p> <p>SCO の使用例:</p> <pre>// set the students response // Student response was to a true-false LMSSetValue("cmi.interactions.0.student_response","f");</pre>
cmi.interactions.n.result	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ :CMIVocabulary (Result) "correct" "wrong" "unanticipated" "neutral" "x.x" (CMIDecimal)</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>定義: 記述された解答に対してシステムがどう判断するか。</p> <p>使用法: これは student_response からの実際の評価結果である。</p> <p>形式: ボキャブラリセット。5 つのとり得るボキャブラリ値。</p> <ul style="list-style-type: none"> ・ correct(正しい) ・ wrong(間違い) ・ unanticipated(予期しない(解答)) ・ neutral(どちらかず) ・ x.x (CMIDecimal) <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる。 ● LMSSetValue(): LMS は以下に従ってエラーコードを設定し、空文字列("")を返すべきである。 <ul style="list-style-type: none"> ・ エラーコード: 401 — 非実装エラー。この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートさ

	<p>れていないことを示すためにエラーコードが設定される。</p> <p>404 — 要素は書き込み専用。SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列("")を返すべきである。</p> <ul style="list-style-type: none"> ● LMSSetValue(): データモデル要素に指定された値を設定する。値は、この要素のデータ・タイプと合致しなければならない。 <ul style="list-style-type: none"> • API 呼び出しの例: LMSSetValue("cmi.interactions.0.result","correct") • 設定値の例: "correct" "95.5" "ananticipated" • エラーコード: 205 — 不正データ・タイプ: LMSSetValue() が起動され、要素に対して設定される値が、不正データ・タイプである。 401 — 非実装エラー。この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを(401)に設定する。 <p>SCO の使用例:</p> <pre>// set the result for the question LMSSetValue("cmi.interactions.0.result","correct");</pre>
cmi.interactions.n.latency	
<p>Supported API calls: LMSSetValue()</p> <p>LMS Mandatory:No</p> <p>データ・タイプ: CMITimespan</p> <p>SCO のアクセス可能性: 書き込み専用</p>	<p>定義: 問題の提示から測定できる反応の完了までの時間。</p> <p>使用法: 反応時間が記録される場合、各レスポンスに対する反応時間値が存在する。</p> <p>形式: コロンによって区切られた時、分および秒。 HHHH:MM:SS.SS</p> <p>時間は最低 2 つの数字および最高 4 つの数字を持つ。分はちょうど 2 つの数字から成る。秒は、2 つの数字とオプションとして小数点と 1 つあるいは 2 つの追加数字を持つ。(つまり 34.45 のように)。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ● 初期化: 値は SCO によってコントロールされる。 ● LMSGetValue(): LMS は以下に従ってエラーコードを設定し、空文字列("")を返すべきである。 <ul style="list-style-type: none"> • エラーコード:

	<p>401 — 非実装エラー. この要素がサポートされないならば、空文字列が返され、そしてこの要素がサポートされていないことを示すためにエラーコードが設定される.</p> <p>404 — 要素は書き込み専用. SCO がこの要素に対して LMSGetValue() を呼び出そうとする場合、LMS はエラーコードに 404 を設定し、空文字列("")を返すべきである.</p> <ul style="list-style-type: none"> ● LMSSetValue(): データモデル要素に指定された値を設定する. 値は、この要素のデータ・タイプと合致しなければならない. <ul style="list-style-type: none"> • API 呼び出しの例: LMSSetValue("cmi.interactions.0.latency","00:29:00") • 設定値の例: "00:29:00" "1234:44:30" • エラーコード: <ul style="list-style-type: none"> 205 — 不正データ・タイプ: LMSSetValue() が起動され、要素に対して設定される値が、不正データ・タイプである. 401 — 非実装エラー. この要素がサポートされないならば、LMS はこの要素がサポートされていないことを示すためにエラーコードを(401 に)設定する. <p>SCO の使用例:</p> <pre>// set the latency for the interaction LMSSetValue("cmi.interactions.0.latency","00:20:00");</pre>
--	--

3.4.5. データタイプおよび制御ボキャブラリ

AICC CMI データモデル 4 の各要素に対するデータ型定義が存在する。以下のセクションでは、データモデル要素に対して指定される各データタイプの仕様を定義する。これらの定義は、さらに API およびデータモデルがどのように実装されるべきかを定義する。

以下の定義は、各データ要素のフォーマットを記述するために使用されるデータタイプである。すべてのデータ・タイプは、それらが AICC CMI データモデル 4 に固有なデータタイプであることを明確に示すために最初に “CMI” の 3 文字を有する。

CMIBlank	空文字列(“”).
CMIBoolean	2 つのボキャブラリ. (“true” または “false”).
CMIDecimal	数値で、小数点を有する場合もある。マイナス記号が前に付かない場合、数は正数とみなされる。例は「2」, 「2.2」および「-2.2」である。

CMIFeedback	<p>インタラクション中の学習者応答の構造化した記述。フィードバックの構造および内容はインタラクションのタイプに依存する。現在定義されているインタラクションは次のとおりである。</p> <p>True-false(○×): フィードバックは次の単一文字のうちのいずれかである。 “0”, “1”, “t”, または “f”.</p> <p>choice(多肢選択): フィードバックはコンマで区切られた 1 つ以上の単一文字である。正しい文字は “0” ～ “9” と “a” ～ “z” である。正しいフィードバックとなるために、すべての文字を選ばなければならない場合、コンマで区切られたリストを{ }括弧で囲まなければならない。</p> <p>fill-in(穴埋め): 長さ 255 文字までの任意の英数字文字列。最初の文字以降に表れる空白は意味を持つ。</p> <p>Numeric(数値): CMIDecimal</p> <p>Likert(アンケート): 単一文字。正しいな文字は 0 から 9 と a から z である。</p> <p>Matching(組合せ): 1 つ以上の識別子のペア。各識別子は単一の文字か、数(0 から 9, a から z)である。ペア中の識別子はピリオドによって区切られる。コンマでペアを区切る。正しいインタラクションとなるために、すべてのペアが正しく組み合わせられなければならない場合、コンマで区切られたペアのリストを{ }括弧で囲まなければならない。</p> <p>Performance(パフォーマンス): これは非常に柔軟性のあるフォーマットである。本質的に 255 文字以下の英数字文字列である。</p> <p>Sequencing(順序付け): コンマで区切られた一連の単一文字。正しい文字は 0～9 と a～z である。文字の順序は、フィードバックの正しさを決定する。</p>
CMIIentifier	中に余白あるいは印刷不可能な文字のない英数字のグループ。最大 255 文字。
CMInteger	0 から 65536 までの整数。
CMISInteger	-32768 から+32768 までの符号付き整数。
CMISString255	最大長が 255 文字の ASCII 文字セット。
CMISString4096	最大長が 4096 文字の ASCII 文字セット。

CMTime	<p>2 4 時間表示での時間表示, 時, 分, 秒で HH : MM : SS.SS の形式で, 指定される.</p> <p>時間と分は, ちょうど 2 桁である. 秒は 2 桁の数とオプションとしての小数点及び 1 桁または 2 桁の追加の数である (すなわち 34.45 のように).</p>
CMTimespan	<p>時間の長さを時, 分, 秒で以下の数値形式で示したもの : HHHH:MM:SS.SS.</p> <p>時間は最低 2 桁および最高 4 桁である. 分はちょうど 2 桁である. 秒は 2 桁の数とオプションとしての小数点及び 1 桁または 2 桁の追加の数である (すなわち 34.45 のように).</p>
CMIVocabulary	<p>スキーマの文脈内に特別なボキャブラリを付加するために使用される. ボキャブラリは, 以下の表に示すボキャブラリと完全に合致しなければならない. 単一文字と略語は, API コミュニケーションでは使用してはならない. 有効なボキャブラリのリストに関しては下記の表を参照されたい.</p>

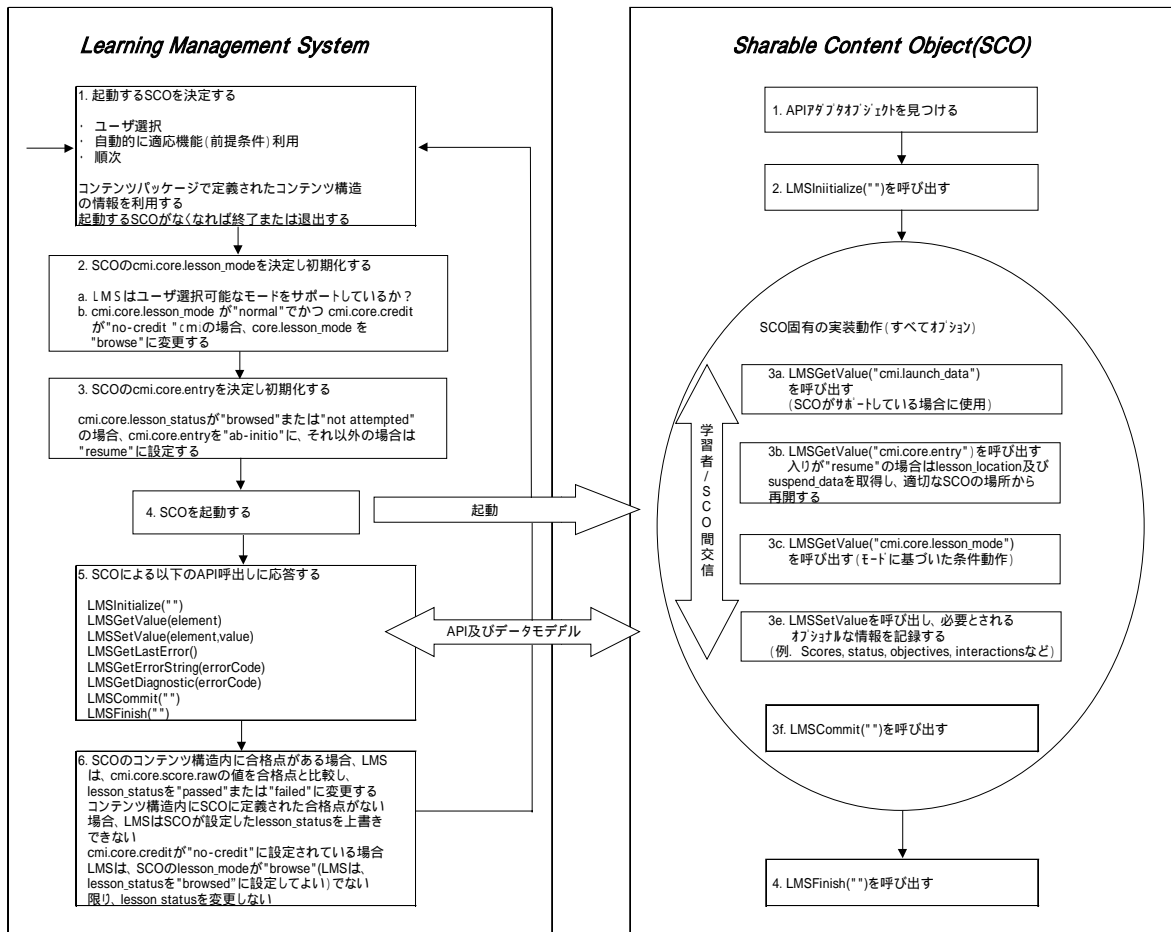
データ・タイプに加えて, いくつかのデータモデル要素は, 結び付けられた可能な値のボキャブラリで定義される. 下記のテーブルは, これらの要素のためのボキャブラリタイプと値を要約したものである (AICC CMI001 Guidelines for Interoperability⁴ による).

ボキャブラリタイプ	ボキャブラリ一覧
Mode/モード cmi.core.lesson_mode	"normal" "review" "browse"
Status/状態 cmi.core.lesson_status cmi.objectives.n.status	"passed" "completed" "failed" "incomplete" "browsed" "not attempted"
Exit/終了 cmi.core.exit	"time-out" "suspend" "logout" "" - (<i>empty string</i>)
Credit/単位取得 cmi.core.credit	"credit" "no-credit"
Entry/開始 cmi.core.entry	"ab-initio" "resume" "" - (<i>empty string</i>)

Interaction/インタラクション cmi.interactions.n.type	"true-false" "choice" "fill-in" "matching" "performance" "likert" "sequencing" "unique" "numeric"
Result/ (解答) 結果 cmi.interactions.n.result	"correct" "wrong" "unanticipated" "neutral" X.X (CMIDecimal)
Time Limit Action/タイム・リミット・アクション cmi.student_data.time_limit_action	"exit,message" "exit,no message" "continue,message" "continue,no message"

3.5. ランタイム環境の動作

次の図は、LMS と SCO(Sharable Content Object)間での可能な実行時の通信シナリオの一例を示すものである。以下の図表は LMS と SCO が描く一般的な実行時動作を示している。ただし、これは単に一例であり、実装要求ではないことに注意されたい。



APPENDIX A

略語リスト

略語リスト

ADL	Advanced Distributed Learning
AICC	Aviation Industry CBT Committee
API	Application Program Interface
ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
ASCII	American Standard Code for Information Interchange
AU	Assignable Unit
AWT	Abstract Window Toolkit
CBI	Computer-Based Instruction
CBT	Computer-Based Training
CDATA	Character Data
CMI	Computer Managed Instructions
COTS	Commercial Off-The-Shelf
CSF	Content Structure Format
DC	Dublin Core
DoD	Department of Defense
DOL	Department of Labor
DTD	Document Type Definition
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDA	Institute for Defense Analyses
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITS	Intelligent Tutoring Systems
LMS	Learning Management System
LOM	Learning Objects Metadata

LTSC	Learning Technology Standards Committee
MIME	Multipurpose Internet Mail Extensions
NGB	National Guard Bureau
OSTP	Office of Science and Technology Policy
PCDATA	Parsable Character Data
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
URI	Universal Resource Identifier
URL	Universal Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language

APPENDIX B

参考文献

参考文献

1. Aviation Industry CBT (Computer-Based Training) Committee.
(<http://www.aicc.org/>)
2. Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC). (<http://ltsc.ieee.org/>)
3. IMS Global Learning Consortium, Inc. (<http://www.imsglobal.org/>)
4. AICC/CMI CMI001 Guidelines for Interoperability Version 3.4. October 23, 2000. Includes: AICC Course Structure Format, AICC CMI Data Model
Available at: <http://www.aicc.org/>.
5. ADL Co-Laboratories. (<http://www.adlnet.org/>)
6. Institute for Defense Analyses (IDA). (<http://www.ida.org/>)
7. Executive Order 13111: Using Technology To Improve Training Opportunities for Federal Government Employees.
8. Gettinger, M. (1984) Individual differences in time needed for learning: A review of the literature. *Educational Psychologist*, 19,15-29.
9. Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31, 104-137.
10. Bloom, B.S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4-16.
11. Fletcher, J. D. (2001) Evidence for Learning from Technology-Assisted Instruction. In H. F. O'Neil Jr. and R. Perez (Eds.) *Technology Applications in Education: A Learning View*. Hillsdale, NJ: Lawrence Erlbaum Associates.
12. Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE). (<http://www.ariadne-eu.org/>)
13. Gibbons, A.S. & Fairweather, P.G. Computer-based Instruction. (2000) In, S. Tobias and J.D. Fletcher (Eds.), Training and Retraining: A Handbook for Business, Industry, Government, and the Military. New York: Macmillan Gale Group.
14. Suppes, P. (1964) Modern learning theory and the elementary-school curriculum. *American Educational Research Journal*, 1, 79-93.
15. Carbonell, J. R., "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," *IEEE Transactions on Man-Machine Systems*, Vol. 11, 1970, pp. 190-202.

-
16. Sleeman, D. & Brown, J. S. (Eds.) (1982) Intelligent Tutoring Systems. New York, NY: Academic Press, 1982.
 17. Woolf, B.P., & Regian, J.W. (2000). Knowledge-based training systems and the engineering of instruction. In S. Tobias and J. D. Fletcher (Eds.), Training and retraining: A handbook for business, industry, government, and the military (339-356). New York: Macmillan Reference.
 18. Gibbons, A.S. & Fairweather, P.G. (1998) Computer-based Instruction: Design and Development. Englewood-Cliffs, NJ: Educational Technology Publications.
 19. Gibbons, A.S. & Fairweather, P.G. (2000) op. cit.
 20. IMS Content Packaging Specification Version 1.1.2.
Available at: <http://www.imsglobal.org/>.
 21. IEEE Information Technology - Learning Technology - Learning Objects Metadata LOM: Working Draft 6.1 (2001-04-18).
As referenced by the IMS Learning Resource Meta-data Specification Version 1.2.
Available at: <http://ltsc.ieee.org/>.
 22. IMS Learning Resource Meta-data Specification Version 1.2.
Includes: IMS Learning Resource Meta-data Information Model, IMS Learning Resource Meta-data XML Binding Specification, and IMS Learning Resource Meta-data Best Practice and Implementation Guide
Available at: <http://www.imsglobal.org/>.
 23. ISO 639: This is an international standard for the representation of languages. Version 1 uses two-letter language codes, e.g. 'en' for English, 'fr' for French, 'nl' for Dutch, etc. These language codes are a basis for the IETF registry of language tags, as specified in RFC 1766: Tags for the identification of languages.
Available at: <http://www.iso.ch/>.
 24. ISO 3166: This is an international standard for the representation of country names, e.g. 'BE' for Belgium, 'CA' for Canada, 'FR' for France, 'GB' for United Kingdom, 'US' for United States, etc.
Available at: <http://www.iso.ch/>.
 25. vCard: This standard defines how contact details for people and organizations can be represented.
Available at: <http://www.imc.org/pdi/>.
 26. ISO 8601: This is an international standard that specifies numeric representations of date and time.
Available at: <http://www.iso.ch/>.
 27. World Wide Web Consortium (W3C). <http://www.w3c.org/>
Includes: Universal Resource Locator, Universal Resource Identifier, Extensible Markup Language Version 1.0, Document Object Model (DOM) Specification.
 28. Dublin Core Metadata Initiative. <http://www.dublincore.org/>.

APPENDIX C

改訂履歷

改訂履歴

SCORM Version	Release Date	Description of Change
1.1	16-Jan-2001	<p>ランタイム環境の章の各節をより詳細化</p> <p>起動</p> <p>API</p> <p>データモデル</p> <p>ランタイム環境の起動の変更</p> <p>SCO の起動をより詳細化</p> <p>ランタイム環境 API の変更</p> <p>LMSInitialize(""), LMSFinish("") 及び LMSCommit("") のパラメータは、現在はすべて空文字列でなければならない。</p> <p>LMSInitialize(""), LMSFinish(""), LMSCommit("") 及び LMSSetValue() は、現在 CMIBoolean ("true" or "false") に変換される文字列を返さねばならない。</p> <p>各 API 呼び出しの記述をより詳細化</p> <p>API エラーコードの使用法の節を追加</p> <p>どのような場合に各エラーコードを使用するかシナリオを定義</p> <p>新しいエラーコードの追加</p> <p>兄弟フレームを探すことのないように API() 検出アルゴリズムを変更</p> <p>ランタイム環境データモデルの変更</p> <p>すべてのデータモデル要素を詳細に記述するための表を追加</p> <p>ランタイムデータモデル要素の変更については e 表 C-1a を参照</p> <p>ランタイムデータモデルタイプの変更については表 C-1b を参照</p> <p>ランタイムデータモデルボキャブラリの変更については表 C-1c を参照</p>
1.2	01-Oct-2001	<p>SCO とアセットが起動されることを示すように図 3.1 a を変更</p>

		起動の節に SCO 以外についての詳細記述を追加
		SCO /LMS 間通信 API の状態遷移と説明を追加
		エラーコードの使用場面を示す API エラーコード使用表の詳細を追加
		<p>以下のデータモデル要素についてより明確化:</p> <p>cmi.core.lesson_status</p> <p>cmi.core.entry</p> <p>cmi.core.exit</p> <p>cmi.core.session_time</p> <p>得点 (cmi.core.score.raw, cmi.core.score.min, cmi.core.score.max, cmi.objectives.n.score.raw, cmi.objectives.n.score.min, cmi.objectives.n.score.max)を表すこれらの要素のより詳細な定義を追加. 以下の文を追加: 「正規化された 0 から 100 までの値でなければならない」</p> <p>全般に表の形式を整形</p>
		3.5 ランタイム環境の動作 を追加
		全般に文法を整形

Table C-1a: Run-Time Environment Data Model Element Changes

Data Model Element	Change from SCORM 1.0
cmi.core._children	データタイプ CMISString256 を CMISString255 に変更
cmi.core.student_id	変更なし
cmi.core.student_name	データタイプ CMISString256 を CMISString255 に変更
cmi.core.lesson_location	データタイプ CMISString256 を CMISString255 に変更
cmi.core.credit	変更なし
cmi.core.lesson_status	変更なし
cmi.core.entry	変更なし
cmi.core.score._children	データタイプ CMISString256 を CMISString255 に変更
cmi.core.score.raw	変更なし
cmi.core.score.min	変更なし
cmi.core.score.max	変更なし
cmi.core.total_time	変更なし

cmi.core.lesson_mode	変更なし
cmi.core.exit	変更なし
cmi.core.session_time	変更なし
cmi.suspend_data	変更なし
cmi.launch_data	変更なし
cmi.comments	この要素は現在は SCO/学習者からのコメントを保持するために使用される SCO はこの要素について LMSGetValue() 及び LMSSetValue() を呼び出せる
cmi.comments_from_lms	この新要素は LMS ないしインストラクターからのコメントを置いて SCO によって使用されるように追加された
cmi.objectives._count	変更なし
cmi.objectives._children	データタイプ CMISString256 を CMISString255 に変更
cmi.objectives.n.id	変更なし
cmi.objectives.n.score._children	scores を score に変更 データタイプ CMISString256 を CMISString255 に変更.
cmi.objectives.n.score._count	学習目標の得点リストのサポートを削除
cmi.objectives.n.score.raw	得点に対するリストのサポートを削除 各学習目標は、現在は一つの得点のみを持つことができる 名前を scores から score に変更.
cmi.objectives.n.score.min	得点に対するリストのサポートを削除 各学習目標は、現在は一つの得点のみを持つことができる 名前を scores から score に変更
cmi.objectives.n.score.max	得点に対するリストのサポートを削除 各学習目標は、現在は一つの得点のみを持つことができる 名前を scores から score に変更.
cmi.objectives.n.status	得点に対するリストのサポートを削除 各学習目標は、現在は一つの得点のみを持つことができる 名前を scores から score に変更.
cmi.objectives.n.mastery_time	削除
cmi.evaluation._children	削除
cmi.evaluation.course_id	削除
cmi.evaluation.comments	削除

emi.evaluation.comments.n.time	削除
emi.evaluation.comments.n.location	削除
emi.evaluation.comments.n.content	削除
emi.evaluation.interactions._children	削除
emi.evaluation.objectives._status._children	削除
emi.evaluation.paths._children	削除
emi.evaluation.performance._children	削除
emi.evaluation.lesson_id	削除
emi.evaluation.date	削除
cmi.interactions._children	追加
cmi.interactions._count	追加
cmi.interactions.n.id	データタイプ CMString256 を CMString255 に変更
cmi.interactions.n.objectives._count	追加
cmi.interactions.n.objectives.n.id	データタイプ CMString256 を CMString255 に変更
cmi.interactions.n.time	変更なし
cmi.interactions.n.type	変更なし
cmi.interactions.n.correct_responses._count	追加
cmi.interactions.n.correct_responses.n.pattern	変更なし
cmi.interactions.n.weighting	変更なし
cmi.interactions.n.student_response	変更なし
cmi.interactions.n.result	変更なし
cmi.interactions.n.latency	変更なし
cmi.student_data._children	データタイプ CMString256 を CMString255 に変更
emi.student_data.attempt_number	削除
cmi.student_data.mastery_score	変更なし
cmi.student_data.max_time_allowed	変更なし
cmi.student_data.time_limit_action	変更なし
emi.student_data.attempt_records._count	削除
emi.student_data.attempt_records._children	削除
emi.student_data.attempt_records.n.lesson_score._children	削除
emi.student_data.attempt_records.n.lesson_score.raw	削除
emi.student_data.attempt_records.n.lesson_score.min	削除
emi.student_data.attempt_records.n.lesson_score.max	削除
emi.student_data.attempt_records.n.lesson_status	削除
emi.student_data.tries_during_lesson	削除
emi.student_data.tries.n.score.raw	削除
emi.student_data.tries.n.score.max	削除
emi.student_data.tries.n.score.min	削除
emi.student_data.tries.n.status	削除
emi.student_data.tries.n.time	削除
emi.student_demographics._children	削除
emi.student_demographics.city	削除
emi.student_demographics.class	削除
emi.student_demographics.company	削除
emi.student_demographics.country	削除
emi.student_demographics.experience	削除

emi.student_demographics.familiar_name	削除
emi.student_demographics.instructor_name	削除
emi.student_demographics.title	削除
emi.student_demographics.native_language	削除
emi.student_demographics.state	削除
emi.student_demographics.street_address	削除
emi.student_demographics.telephone	削除
emi.student_demographics.years_experience	削除
cmi.student_preference._children	データタイプ CMString256 を CMString255 に変更
cmi.student_preference.audio	変更なし
cmi.student_preference.language	データタイプの CMString256 を CMString255 に変更
emi.student_preference.lesson_type	削除
cmi.student_preference.speed	変更なし
cmi.student_preference.text	変更なし
emi.student_preference.text_color	削除
emi.student_preference.text_location	削除
emi.student_preference.text_size	削除
emi.student_preference.video	削除
emi.student_preference.windows.n	削除
emi.paths.n.location_id	削除
emi.paths.n.time	削除
emi.paths.n.status	削除
emi.paths.n.why_left	削除
emi.paths.n.time_in_element	削除

Table C-1b: Run-Time Environment Data Type Changes

CMIDate	削除
CMIBoolean	型をすべて小文字に変更 (“true”, “false”)
CMIFeedback	CMIFeedback データタイプの使用法を定義するため詳細情報を追加
CMILocale	削除
CMISIdentifier	削除
CMString255	CMString255 の最大長を 255 文字に変更
CMITime	オプションである小数点以下の数字を最大 2 桁までに (i.e. 34.43)
CMITimespan	オプションである小数点以下の数字を最大 2 桁までに (i.e. 34.43)

Table C-1c: Run-Time Environment CMIVocabulary Changes

Exit	ボキャブラリ “” (空文字列) を追加
Why Left	削除
Credit	“no credit” を “no-credit”に変更
Entry	ボキャブラリ “” (空文字列) を追加
Time Limit Action	すべてのボキャブラリ選択を変更
Interaction	“multiple choice” を “choice”に変更 “fill in the blank” を “fill-in”に変更 “simple performance” を “performance”に変更

