IMS Learner Information Package Accessibility for LIP Conformance Specification

Version 1.0 Final Specification

Document Name: IMS Learner Information Package Accessibility for LIP Conformance Specification
Date: 18 June 2003

Table of Contents

1. Introduction

This document describes a conformance specification for Accessibility for LIP (ACCLIP). ACCLIP is a set of name spaced extensions to the IMS Learner Information Package.

1.1 Overview

This conformance plan is, by necessity, a high level abstract plan, since actual conformance testing must be done within the context of an application profile. As such, this document can be viewed as source material for developing specific ACCLIP test plans tightly associated with an application profile.

This document draws heavily on the ACCLIP Information Model and ACCLIP Object Model. The reader is referred to those documents for the definitive specification of data elements, methods, and class definitions.

1.2 Scope and Context

A general conformance plan is described for the <accessForAll> and <accommodation> elements as defined by the ACCLIP Information Model. Object model tests are described for each of the major classes defined by the ACCLIP Object Model. These objects do not go all the way to specific preferences, since they are implementation dependent.

1.3 Nomenclature

The following abbreviations and acronyms are used in this document.

ACCLIP	Accessibility for Learner Information Package
LIP	Learner Information Package
XML	Extensible Mark-up Language

1.4 References

The following document references are identified:

[ACCLIP 1a,b,c,d]	IMS Learner Information Package Accessibility for LIP Information Model, Binding, Best Practice Guide, and Use Cases
[LIP 1a,b,c]	IMS Learner Information Package Information Model, Binding, and Best Practice Guide

2. The Conformance Architecture

2.1 Use Case Coverage

The IMS Accessibility for LIP Use Cases, v1.0 describes several likely use cases for accessibility preferences. Several of these use cases describe actual distance learning environments that utilize accessibility preferences. All of the use cases described refer to preferences and accommodations that can be represented using an ACCLIP structure.

2.2 The System Architecture

The ACCLIP specification does not presume a particular architecture beyond that specified in the IMS Abstract Framework. Since ACCLIP is intended to provide accessibility preferences in a comprehensive learner profile, it is likely that an ACCLIP manager is part of a larger Learner Profile Manager. This is not, however, a requirement. The ACCLIP preferences can be implemented and used in isolation and do not have dependencies on the Learner Information Package beyond those stated in the ACCLIP Information Model.

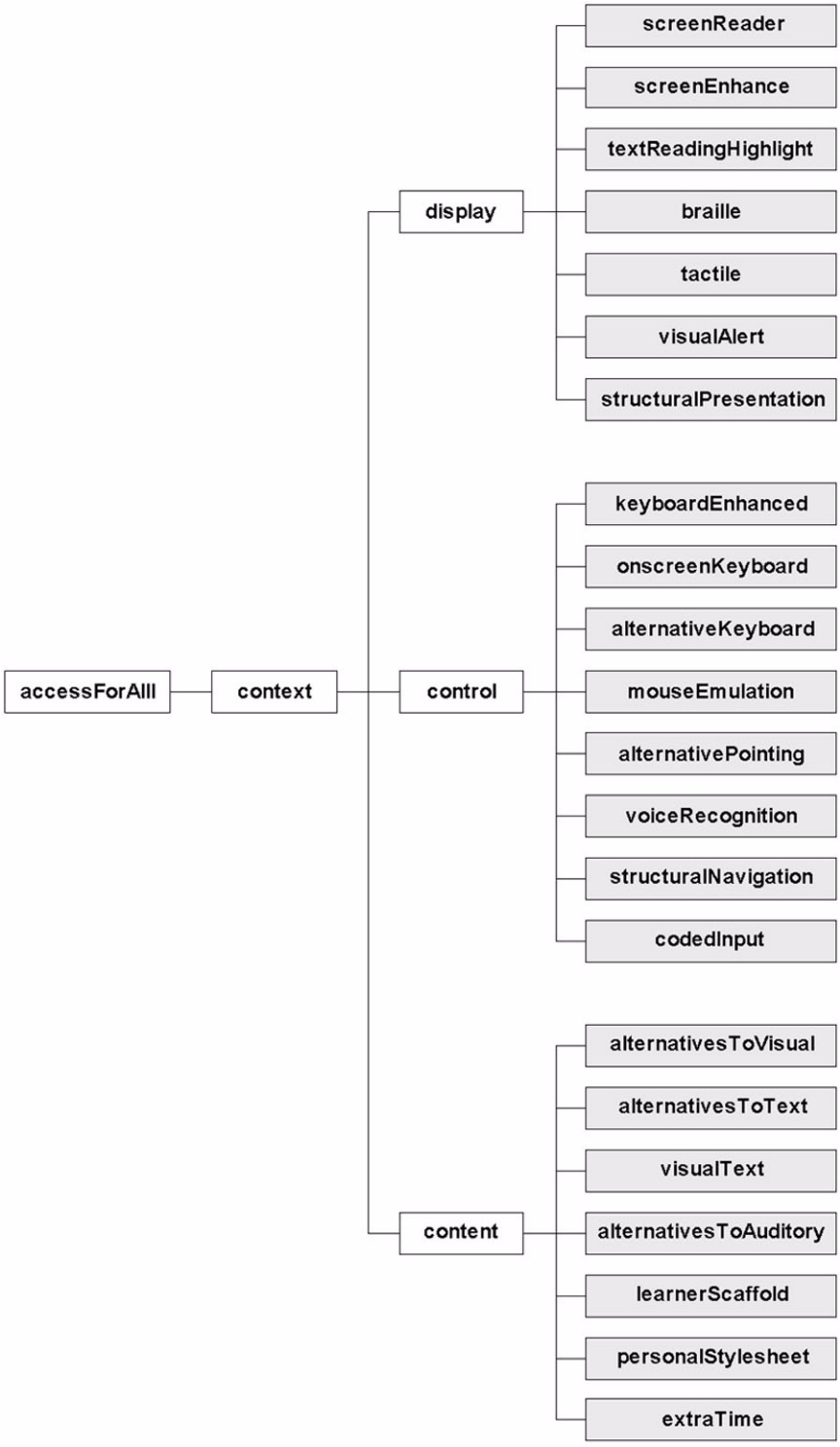
Regardless of how ACCLIP is implemented, two classes of systems are identified which must interact with ACCLIP data: ACCLIP repositories and ACCLIP applications. ACCLIP repositories are responsible for creating, storing, and managing ACCLIP data. ACCLIP applications allow preferences to be utilized. In some cases, an ACCLIP application may alter or add new preferences (a volume change, for example). In such cases, the application supplies new information to a central manager for storage. These two classes of systems are reflected in two distinct conformance classifications: manager and application.

2.3 The Specification Classes

Two root classes are defined in the ACCLIP Information Model: <accessForAll> and <accommodation>. Both are name spaced extensions of elements previously defined in the IMS Learner Information Package.

2.3.1 <accessForAll> Specification Classes

Classes associated with <accessForAll> are defined by the ACCLIP Object Model section of the ACCLIP Information Model, v1.0.



Conformance class implementation is optional depending on the two classes of conformance:

- M – Mandatory Implementation
- O – Optional Implementation

Class	Manager Implementation	Application Implementation
accessForAll	M	M
context	M	M
display	M	O
screenReader	M	O
screenEnhance	M	O
textReadingHighlight	M	O
braille	M	O
tactile	M	O
visualAlert	M	O
structurePresentation	M	O
control	M	O
keyboardEnhanced	M	O
onscreenKeyboard	M	O
alternativeKeyboard	M	O
mouseEmulation	M	O
alternativePointer	M	O
voiceRecognition	M	O
structuralNavigation	M	O
codedInput	M	O
content	M	O
alternativesToVisual	M	O
alternativesToText	M	O
visualText	M	O
alternativesToAuditory	M	O
learnerScaffold	M	O
personalStylesheet	M	O
extraTime	M	O

2.3.2 <accommodation> Specification Classes

Classes associated with <accommodation> are defined by the ACCLIP Object Model section of the ACCLIP Information Model, v1.0.

Conformance class implementation is optional depending on the two classes of conformance:

- M – Mandatory Implementation
- O – Optional Implementation

Class	Manager Implementation	Application Implementation
Accommodation	M	O

2.4 Conformance Test Architecture

The <accessForAll> object model describes three separate access methods for ACCLIP preferences. These are:

1) preference aggregation using XML
2) preference aggregation using objects
3) specific preference references using path descriptions

Each of these are binding dependent and will require separate test methods to measure conformance. In addition to the above, LIP data may be aggregated into an IMS Content Package and, by extension, ACCLIP data as well.

3. accessForAll Test Specification

3.1 <accessForAll> Class

3.1.1 Class Overview

The <accessForAll> class serves as the root data object for ACCLIP preferences. This class serves as a container for context objects.

3.1.2 XML Behavior Specification

The read() method associated with this class returns an XML string formatted according to an accessForAll XML schema binding. Similarly, the write() method allows aggregated preferences to be saved all at once.

3.1.3 Object Behavior Specification

The <accessForAll> class contains a single sub-object called <context>. Methods are provided for getActiveContext(), setActiveContext(), getContext(id), and setContext(id).

3.1.4 Path Behavior Specification

Not applicable.

3.2 <context> Class

3.2.1 Class Overview

The <context> class provides a method to group preferences according to need, environment, and alternative use. Each <context> is identified with a identifier unique to this system. The <context> class serves as a container for the <display>, <control>, and <content> objects.

3.2.2 XML Behavior Specification

The read() method associated with this class returns an XML string formatted according to an accessForAll XML schema binding for this context. Similarly, the write() method allows aggregated preferences to be saved all at once in the specified context.

3.2.3 Object Behavior Specification

The <context> class contains three sub-objects and provides get and set methods for each of them. These include getDisplay(), setDisplay(), getControl(), setControl(), getContent(), and setContent(). In addition, two methods are provided to getLang() and setLang(), which is the default language associated with this context.

3.2.4 Path Behavior Specification

Not applicable.

3.3 <display> Class

3.3.1 Class Overview

The <display> class contains learner preferences that determine how information displayed or delivered to the learner. This class serves as a container for the screenReader, screenEnhance, textReadingHighlight, braille, tactile, visualAlert, and structurePresentation objects.

3.3.2 XML Behavior Specification

The read() method associated with this class returns an XML string formatted according to an accessForAll XML schema binding for the display preferences in this context. Similarly, the write() method allows aggregated display preferences to be saved all at once.

3.3.3 Object Behavior Specification

The <display> class contains seven sub-objects and provides get and set methods for each of them. These include getScreenReader(), setScreenReader(), getScreenEnhance(), setScreenEnhance(), getTextReadingHighlight(), setTextReadingHighlight(), getBraille(), setBraille(), getTactile(), setTactile(), getVisualAlert(), setVisualAlert(), getStructurePresentation(), and setStructuralPresentation(). The actual implementations of these sub-objects is dependent on an application profile suitable for its use.

3.3.4 Path Behavior Specification

The getViaPath(path) method resolves a path specification to a specific display preference and returns it. The setViaPath(path, value) method resolves a path specification and changes the value associated with it.

3.4 <control> Class

3.4.1 Class Overview

The <control> class contains learner preferences that determine learners interact with an eLearningsystem. This class serves as a container for the <keyboardEnhanced>, <onscreenKeyboard>, <alternativeKeyboard>, <mouseEmulation>, <alternativePointer>, <voiceRecognition>, <structuralNavigation>, and <codedInput> objects.

3.4.2 XML Behavior Specification

The read() method associated with this class returns an XML string formatted according to an accessForAll XML schema binding for the control preferences in this context. Similarly, the write() method allows aggregated control preferences to be saved all at once.

3.4.3 Object Behavior Specification

The <display> class contains seven sub-objects and provides get and set methods for each of them. These include getKeyboardEnhanced(), setKeyboardEnhanced(), getOnscreenKeyboard(), setOnscreenKeyboard(), getAlternativeKeyboard(), setAlternativeKeybaord(), getMouseEmulation(), setMouseEmulation(), getAlternativePointer(), setAlternativePointer(), getVoiceRecognition(), setVoiceRecognition(), getStructuralNavigation(), setStructuralNavigation(), getCodedInput(), and setCodedInput(). The actual implementations of these sub-objects is dependent on an application profile suitable for its use.

3.4.4 Path Behavior Specification

The getViaPath(path) method resolves a path specification to a specific control preference and returns it. The setViaPath(path, value) method resolves a path specification and changes the value associated with it.

3.5 <content> Class

3.5.1 Class Overview

The <content> class contains learner preferences that define the learner's preferences in content. This class serves as a container for the <alternativesToVisual>, <alternativesToText>, <visualText>, <alternativesToAuditory>, <learnerScaffold>, <personalStylesheet>, and <extraTime> objects.

3.5.2 XML Behavior Specification

The read() method associated with this class returns an XML string formatted according to an accessForAll XML schema binding for the content preferences in this context. Similarly, the write() method allows aggregated content preferences to be saved all at once.

3.5.3 Object Behavior Specification

The display class contains seven sub-objects and provides get and set methods for each of them. These include getAlternativesToVisual(), setAlternativesToVisual(), getAlternativesToText(), setAlternativesToText(), getVisualText(), setVisualText(), getAlternativesToAuditory(), setAlternativesToAuditory(), getLearnerScaffold(), setLearnerScaffold(), getPersonalStylesheet(), setPersonalStylesheet(), getExtraTime(), and setExtraTime. The actual implementations of these sub-objects is dependent on an application profile suitable for its use.

3.5.4 Path Behavior Specification

The getViaPath(path) method resolves a path specification to a specific content preference and returns it. The setViaPath(path, value) method resolves a path specification and changes the value associated with it.

4. Conformance Categories

The ACCLIP Conformance strategy defines two levels of conformance: Manager and Application. Generally speaking, repositories that manage ACCLIP information are required to implement all ACCLIP structures. Applications are required to implement the root element, context, and at least one set of technology preferences.

4.1 Manager Level Conformance

ACCLIP repositories manage ACCLIP information as defined by the IMS Accessibility for LIP Information Model, v1.0.

4.1.1 Baseline Conformance Classification

Not applicable.

4.1.2 Strict Conformance Classification

Manager Level Conformance Classification requires conformance to all ACCLIP elements and data.

4.1.3 Profiled Conformance Classification

Not applicable.

4.2 Application Level Conformance

Applications that use some or all ACCLIP preferences and accommodations are not required to implement all such preferences. They are required to implement the root elements (<accessForAll> and <accommodation>), the <context> element, and at least one set of technology preferences in <display>, <control>, or <content>.

A quick example illustrates why all preferences are optional under this conformance strategy. Screen readers are applications that convert textual information into speech. Several preferences are provided in <screenReader> for speech rate, pitch, volume, etc. Since these preferences are specific to the application, there is not need for the application to implement other preferences, such as braille, etc.

4.2.1 Baseline Conformance Classification

Application Level Conformance requires at a minimum to implement root elements (<accessForAll> and <accommodation>), <context>, and at least one set of technology elements in <display>, <control>, or <content>.

4.2.2 Strict Conformance Classification

Not applicable, since most elements are optional.

4.2.3 Profiled Conformance Classification

Since some applications may be more general than a screen reader (using the example above), groups of technology preferences can be collected into profiles. Three profiles are initially defined: display, control, and content. Applications choosing to conform to the display, control, or content profiles are required to implement all of the technology preferences associated with them. Other mixed profiles may also be defined.

Appendix A – Glossary of Terms

TBD.

Appendix B – Baseline Vocabularies

Element	Token	Restricted
Link		y
	speakLink	
	differentVoice	
	soundEffect	
	none	
Usage		
	required	y
	preferred	
	optionally used	
	not used	
GenericFace		
	serif	n
	sansSerif	
	monospaced	
ReadingUnit		
	word	n
	sentence	
	paragraph	
	continuous	
Grade		
	1	y
	2	
	uncontracted	
	contracted	
StatusCell		y
	off	
	left	
	right	
SystemSounds		y
	none	
	desktop	
	window	
	captionBar	

Element	Token	Restricted
ContextDensity		y
	collapsed	
	expanded	
ContextViews		y
	imageIntensive	
	textIntensive	
WindowLayout		y
	tiled	
	overlap	
	frontMost	
AlphaLayoutInternal		y
	standard	
	sequential	
	frequency	
SwitchType		n
	mouse	
	game	
	serial	
	usb	
	firewire	
	infrared	
	bluetoothe	
SwitchAssignment		y
	select	
	cancel	
	scan	
	right	
	left	
	up	
	down	
	horizontal	
	vertical	
Handedness		y
	left	
	right	

Element	Token	Restricted
Vocabulary		y
	context	
	natural	
NavigationDepth		y
	depthFirst	
	breadthFirst	
Code		y
	morse	
	quartering	
	eightCell	
	chordic	
Codetermination		y
	switch	
	timed	
CodeSelection		y
	pointAndClick	
	pointAndDwell	
AudioDescription		y
	standard	
	expanded	
SignLanguage		n
	American-ASL	
	Australian- Auslan	
	Austrian-ASQ	
	British-BSL	
	Danish-DSL	
	French-LSF	
	German-DGS	
	Irish-ISL	
	Italian-LIS	
	Japanese-JSL	
	Malaysian-MSL	
	Mexican-LSM	
	Native-American	
	Netherlands-NGT	

Element	Token	Restricted
	Norwegian-NSL	
	Quebec-LSQ	
	Russian-RSL	
	Singapore-SLS	
	Spanish-LSE	
	Swedish-SWL	
	other	
LearnerScaffold		n
	dictionary	
	calculator	
	noteTaking	

Appendix C – Conformance for Extensions

The <display>, <control>, and <content> elements of <accessForAll> each have a <futureTechnology> element that is the preferred method for adding preference extensions.

About This Document

Title	IMS Learner Information Package Accessibility for LIP Conformance Specification
Editor	Mark Norton
Team Co-Lead	Jutta Treviranus
Version	1.0
Version Date	18 June 2003
Status	**Final Specification**
Summary	This document describes the ACCLIP Conformance.
Revision Information	18 June 2003
Purpose	Defines conformance measures for the ACCLIP specification.
Document Location	http://www.imsglobal.org/acclip/acclipv1p0/imsacclip_confv1p0.html

List of Contributors

The following individuals contributed to the development of this document:

Name	Organization
Cathleen Barstow	The CBP/WGBH National Center for Accessible Media
Anastasia Cheetham	University of Toronto, ATRC, Industry Canada
Martyn Cooper	Open University, UK
Eric Hansen	Educational Testing Services
Andy Heath	UK eUniversities Worldwide, Sheffield Hallam University
Phill Jenkins	IBM
Hazel Kennedy	Open University, UK
Liddy Nevile	IMS Australia
Mark Norton	IMS Global Learning Consortium, Inc.
Madeleine Rothberg	The CBP/WGBH National Center for Accessible Media
Joseph Scheuhammer	University of Toronto, ATRC, Industry Canada
Brendon Towle	Thomson NETg
Jutta Treviranus	University of Toronto, ATRC, Industry Canada
David Weinkauf	University of Toronto, ATRC, Industry Canada

Revision History

Version No.	Release Date	Comments
Final v1.0	18 June 2003	The first public release of the document.

Index