# Technology Enhanced Learning: Conformance - European Requirements & Testing

| | |
|---|---|
| **Title:** | D6 –Acceptance Test Criteria Document |
| **Editor(s):** | Phil Nicholls and Tom Franklin |
| **Version:** | 1.7 |
| **Version date:** | 8 July 2004 |
| **Status:** | Public |
| **Distribution:** | Commission and public |
| **Summary:** | This document describes all acceptance tests that need to be applied to content within the Telcert system. |

# 1 REVISION HISTORY

| Version Number | Release Date | Comments |
|---|---|---|
| D06.0.1 | 7th April 2004 | First Draft |
| D06 v1.0 | 26th April 2004 | Revised after review |
| D06 v1.1 | 4th May 2004 | Revision after project meeting |
| D06 v1.2 | 18th May 2004 | 2 new tests added (generic) |
| D06 v1.3 | 19th May 2004 | Document ordering changed |
| D06 v1.4 | 31 May 2004 | Mostly minor changes to numbering plus addition of introduction etc. |
| D06 v1.5 | 11 June 2004 | Various minor changes |
| D06 v1.6 | 25 June 2004 | Restructuring to move non-Telcert tests to appendix plus minor changes |
| D06 v1.7 | 8 July 2004 | Final edits |

# 2 Executive Summary

In Technology Enhanced Learning, Conformance Testing is a key element in delivering the interoperability required to accelerate market take up of open technologies. Two barriers must be overcome however: 1) Developers are using novel technologies and techniques for which test strategies are not yet available; 2) The growing number of specifications and standards need localisation to meet cultural, pedagogical and organisational needs, and so will the tests.

TELCERT will exploit state-of-the art research to define flexible testing strategies, using new development technologies and techniques which can accommodate localisation, defined as user-community Application Profiles. The initial focus will be on content interoperability testing. The range of specifications available will allow the techniques developed to be readily applied in other Technology Enhanced Learning areas.

This document is a deliverable D06 of the Telcert project and describes the tests which the test suite being developed within the project must undertake to determine whether products are conformant or not. This document should be read in conjunction with the user requirements document.

# 3 TABLE OF CONTENTS

# 4 INTRODUCTION

## 4.1 Scope of the document

This document covers the testing of materials and content covered by the specifications and profiles listed in 6.1 Relevant specifications and standards. The document further outlines the actual tests that should be carried out against each profile and specification, with specified end states, pass/fail criteria and so forth. There is an underlying assumption that this document will be read in conjunction with D5 – Test System Requirements – which documents the requirements criteria of any semi-automated test system. It should be noted that it includes all the tests that partners would like to see the TELCERT system perform, including those that are out of scope of the system or are impractical or impossible to perform (such as testing for no errors).

## 4.2 Intended readership

The document is intended as a technical document and should be read primarily by test software developers. High level descriptions of the tests should be of use to content developers that wish to produce content that is conformant to the application profiles listed below.

# 5 LIST OF ACRONYMS

| **Acronym** | **Expansion** |
|---|---|
| ADL | Advanced Distributed Learning initiative |
| API | Application Profile Interface |
| CMI | Computer Managed Instruction |
| FE | Further Education |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| IIS | Internet Information Server |
| LMS | Learning Management System |
| LOM | Learning Object Metadata |
| ODBC | Open Database Connectivity |
| QTI | Question and Test |
| RFC | Request for Comment |
| SCORM | Sharable Content Object Reference Model |
| URL | Uniform Resource Locator |
| WSDL | Web Services Description Language |
| XML | eXtensible Markup Language |
| XSD | XML Schema Definition |

# 6 SCOPE OF TESTING

## 6.1 Relevant specifications and standards

UFI has created application profiles against a number of specifications and these are detailed individually below. However, it is noted that certain elements of the base specifications do not appear in the schema documents that describe them. Where possible, all of these points have been included in this section and also where the tests need to be described.

### 6.1.1 IEEE Learning Object Metadata (LOM)

The LOM Metadata specification defines a number of sections for metadata that are used by UFI when describing materials, courses and modules. Within UFI's namespace, a course is made up of modules, and a module is made up of materials. Courses and modules share a common definition – that is, within LOM the same fields are used to describe both Courses and Modules. Materials are also described within LOM but make use of different fields.

UFI's use of the LOM specification is described in detail within the application profiles for both UFI courses (for courses and modules) and UFI materials. Only the LOM schema definition is used by UFI.

It should be noted that the UFI profiles impose significant restraints on not only the cardinalities of the various elements, but also on the sizes of some of the fields that are used.

### 6.1.2 IMS Content Packaging version 1.1.1

The IMS Content packaging specification defines a number of conditions that are relevant to conformance and these are upheld by UFI's view. The Content Packaging Conceptual model defines a number of key elements, some of which form the schema and are thus subject to profiling. Some key elements and their constraints do not exist within the schema and thus are defined here.

#### 6.1.2.1 Package Interchange File

This is a single file which includes a top level manifest, and all of the physical files that are identified within that manifest. The IMS identifies that a typical manifest might be for example a .zip, .cab or .jar file.  They state any .zip file must conform to RFC 1951, and it further describes PKZip v2.04g as the recommended default Package Interchange File Format.

For the purposes of testing, UFI has identified that a .zip file that can be decompressed using winzip (version 7.0 or later) or PKZip (version 2.5 or later) is acceptable. IMS requires all content to be capable of being decompressed using the GZIP utility. Content must be capable of being decompressed in this manner to be considered conformant.

Where content is packaged on a CD or other media, and not as a single file, then this test is waived.

#### 6.1.2.2 Top Level Manifest file name

The IMS has identified that a package is valid if the top level manifest exists at the root level of the logical directory, and is named `imsmanifest.XML`. This is true also for conformance with UFI's systems. Any supporting XML files should also be placed at the root level of the package interchange file (or other packaging image).

This manifest file must be present in the correct place, and with the correct name, for content to be considered conformant. The file name must be written in lowercase letters. Supporting files must also be present for content to be considered conformant.

#### 6.1.2.3 Top Level Manifest

This is an XML element and is located in the `imsmanifest.XML` file described above. The application profile defines which elements are used and not used, and should be consulted to determine conformance requirements.

#### 6.1.2.4 Resources

These are the other physical files that make up the package. These files are the actual content that will be executed within UFI's Learning Management System (LMS) environment. Each file must be described within a Manifest to be considered conformant, and no files should exist that are not described within a manifest.

Additionally, any files that are described within a manifest must actually exist; at this time file validation is not considered (i.e., the manifest describes that (for example) a text file exists. That the filename is present within the package is enough at this stage for conformance, that the text file might in reality be an image that has been renamed as a text file is beyond the scope of this document).

### 6.1.3 IMS Learning Resource Meta-data Specification Version 1.2.1

IMS Metadata is used sparingly within the profiles developed by UFI. IMS metadata appears in the content packaging profile, where it is used in the main manifest file to describe the manifest itself. As with the LOM metadata, adherence to the application profile is required for conformance. It is worth noting that the profile itself is strictly a subset of the specification, so all metadata that conforms to the profile is write compliant with the specification. (However the reverse is not true).

### 6.1.4 ADL SCORM Run-Time Environment Version 1.2

The SCORM runtime forms the backbone of the UFI system and it is of vital importance that any content makes valid use of the API as defined in the specification from the ADL. Note that UFI does not use the SCORM content aggregation nor SCORM metadata (although the differences between SCORM metadata and the LOM as used by UFI are almost negligible).

The sections of the runtime specification which are of the greatest importance are those sections that pertain to the API and to the Data Model.

#### 6.1.4.1 API

The SCORM runtime specification presents an eight call API which is used by content to communicate with the LMS. The API is conceptually simple and essentially presents a transaction model, whereby content can initialise a session, read, write and commit data, and finally close the session. In addition a number of methods are made available for the reporting of errors and system diagnostics.

Content is ‘launched’ from a web browser and then has to interrogate its home window and any parent windows until it locates an API instance with which to communicate. The API itself is implemented by the LMS provider and contains proprietary code to communicate with the LMS. Upon locating the API, the Content needs to make a call to the “LMSInitialise” method. Communication then follows, and when the content is closed down or completed, the “LMSFinish” method is called.

Content to be certified needs to make use of this API in the correct manner. Thus content needs to be able to:

- Locate the API
- Make calls on the API
- Handle responses from the API

Content must not attempt to call methods that do not exist within the API. The SCORM runtime specification includes a simple state model showing the sequence of calls (as described above). At a bare minimum, content must conform to the API state model. Content must also conform to the UFI application profile for the SCORM runtime which is covered in a later section.

It should be noted that the API transacts all data in the form of strings. See the following section on the data model for more information.

#### 6.1.4.2 Data Model

The SCORM runtime specification presents a data model which must be used by content when communicating with the server. The model is expressed in a “dot notation” where different attributes are ascribed a path and a value. Although all data in the API is transacted in string format, the underlying data model specifies that certain transacted data values must be parsed into other types. For example, the data model may specify that the type for a value of particular attribute is of type “integer”. In this case, the data that is sent MUST be able to be parsed into an integer value. A value of “23” in this case would be valid; a value of “23A” would not be. Compliant content can only contain attributes that conform to the specified model types.

The data model fully enumerates the valid paths that are permitted for attributes to write data to. Certain of the attributes are read only, and some are write only. Compliant content must only read from readable sources and write to writable destinations. Compliant content must not attempt to write to a read only destination and vice versa. Some attribute paths operate as arrays, whereby the attribute can appear more than once for a particular piece of content. In these cases, integers are found within the dot notation path. Compliant content must only use arrays where the specification dictates that arrays occur.

UFI has profiled the data model that it supports, whereby some of the optional elements in the SCORM runtime specification are not supported. Compliant content must adhere to the application profile developed.

### 6.1.5 IMS Learning Information Package Specification Version 1.0

Learner Information is a collection of information about a Learner (individual or group learners) or a Producer of learning content (creators, providers or vendors). The IMS Learner

Information Package (IMS LIP) specification addresses the interoperability of internet-based Learner Information systems with other systems that support the Internet learning environment. The intent of the specification is to define a set of packages that can be used to import data into and extract data from an IMS compliant Learner Information server. A Learner Information server may exchange data with Learner Delivery systems or with other Learner Information servers. It is the responsibility of the Learner Information server to allow the owner of the learner information to define what part of the learner information can be shared with other systems. The core structures of the IMS LIP are based upon: accessibilities; activities; affiliations; competencies; goals; identifications; interests; qualifications, certifications and licences; relationship; security keys; and transcripts.

IMS LIP forms the basis of the European Diploma Supplement (EDS).

# 7 TARGET PRODUCTS

## 7.1 Product Classification

UFI have identified that the test system should test the following targets:

- Content: This is an executable element which communicates with an LMS. This is typically code that is loaded by the LMS which enables a learner to undertake some online learning. It might, for example, be a piece of Flash that covers a particular topic.

- XML files: These are the files that accompany content, located within the content package. The files to be tested are:

    - Content package manifest file (which is tested both against the profile for content packaging and content packaging metadata).

    - LOM metadata files (for the course/module/material as appropriate).

- Content Packages: As described in section 5, the distribution of the content, be it as a package interchange file or as a collection of files in some other media.

Note that the LMS itself is not considered to be a test target.

Content is typically flash or shockwave content that is run from within a web page on a web server.

## 7.2 Product Variability

This section looks at variability within the products described above. Most notably, this means how the application profiles should be applied to the target of the test. Detailed information on the profiles, and the profiles themselves, can be found in the deliverable D4.

### 7.2.1 Content

Content may be written in a number of different formats, such as static web pages, flash or shockwave. The format of the content is not important as regards testing; as noted earlier it is the interaction between content and LMS that is of importance. All content must conform to the behavioural application profile for API interactions. Further, all data exchanged between content and LMS must adhere to the CMI application profile which outlines the differences between the types used in the SCORM specification and the types used in UFI's implementation.

Content that conforms to the profiles above is deemed to be conformant.

### 7.2.2 XML Files

XML files are used to describe content packages and metadata. A number of files may be tested; according to the XML specification all files will by default be encoded using UTF-8 unless some other encoding is specified in the XML file.

#### 7.2.2.1 Content Packaging Manifest File

This is typically invariant. Manifest files must conform to the application profile for content packaging. Although the specification allows for relatively elastic definitions on the extent of manifests, within the UFI system content typically includes only one manifest file.

#### 7.2.2.2 Metadata files

Metadata is found in two places:

1. The file is located within a content package, and is an XML file describing the material / module or course.

2. The metadata is embedded into the manifest for the content package, and contains metadata on the manifest as a whole.

In both cases, the metadata must conform to the relevant application profile from UFI.

### 7.2.3 Content Packages

Content packages exist in one of two forms:

1. The package is a collection of files on a CD or other media.

2. The package is a single file which is an archive of files.

In both cases, the testing requirements are the same (except for the additional checking of the archive file format in the second option), and conformant content in these cases must validate against the IMS specification.

# 8 PLATFORM REQUIREMENTS

Platform is the combination of hardware, software and network environment. For conformance candidates, the following platform requirements are present.

## 8.1 Hardware

REFERENCED SPECIFICATION: (none)

TEST TYPE: Other

Hardware is not an issue for XML files. Content itself will ultimately be executed from a Solaris, Apple or Intel server. Content Packages must be able to be decompressed onto the server.

# 9 VALIDATION

This section describes the range of tests that will be applied against the target products. Tests are examined for each of the previously identified target types, and criteria established for all valid end states of those tests (pass/fail/untested).

## 9.1 Constraints on Testing

UFI envisages that the TELCERT test system will operate alongside UFI's current testing software. Content developers will be provided with a remote logon to the test system, and this should first be used by them to get their content into a state whereby they are satisfied that it is conformant with UFI's profiles. At this stage, the content is sent to UFI for formal acceptance testing. At this time, the content would be retested on the TELCERT system to ensure the basic conformance tests have been passed. Thus the test system should be available remotely.

## 9.2 Interpretation

Content will be classified as conformant if it passes all of the conformance tests that apply to the application profile that it was developed against. All tests must be performed, and all must be passed. For tests with multiple sub-elements, all sub-elements must be tested and passed.

## 9.3 Consolidated Test List

This list contains the list of tests for all target products, based on application profiles, base specifications, environmental factors and so forth.

### 9.3.1 Content Launch Tests

These tests ensure that content can be launched and that the content is able to communicate with the API.

#### 9.3.1.1 Launch Test

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.2.1

TEST TYPE: Base Specification (not specified in the XSD)

DESCRIPTION: Content must be able to be launched from an LMS, or in this case from the test system. Launching is the process of the user indicating that the content should be commenced.

PASS: The content starts and displays on the screen.

FAIL: The content does not start; does not display on the screen.

#### 9.3.1.2 API Located

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.6.1 / UFI Application Profile.

TEST TYPE: Base Specification – SCORM Runtime (not specified in the XSD)

DESCRIPTION: Successfully launched content must locate an API with which it can interact. The test system should make a suitable API available

PASS: Content locates the API and makes use of it.

FAIL: Content does not locate the API.

#### 9.3.1.3 API Location Fail

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment section 3.3.6.1 / UFI Application Profile

TEST TYPE: Base Specification – SCORM Runtime (not specified in the XSD)

DESCRIPTION: When the LMS is not available, then content should be able to gracefully degrade. For the purposes of this test, the test system should not make an API available for content to use. Users should be made aware that their work will not be sent to the LMS.

PASS: Content can handle the lack of an API, informing the user of the fact.

FAIL: The Content halts execution, or carries on regardless, with no alert message displayed or passed to the user.

### 9.3.2 Content API Tests

These tests cover each of the API calls within the SCORM specification and the UFI profile.

An assumption is made that at this stage, the content has located and is able to use an API. The profile specifies a calling pattern for each call, usually requiring that calls to the error checking methods are followed.

#### 9.3.2.1 Initialise Test

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1, 3.3.2.2 and 3.3.6.1 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

DESCRIPTION: Given a valid API to work with, the content should first call LMSInitialise(“”).

PASS: The first call from the content is LMSInitialise.

FAIL: The first call from the content is not LMSInitialise.

#### 9.3.2.2 Initialise Test – Responses

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.2.1 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

DESCRIPTION: Given a call from the content to LMSInitialise(“”), the content should be able to handle all of the responses below.

- Response = “true”
  PASS: Content can handle the response; go to test 9.3.2.7
  FAIL: Content reports an error, or closes down for some reason.

- Response = “false”
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

- Response = “anything that is not in the specification” / no response
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

#### 9.3.2.3 Get Test – Responses

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.2.1 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

DESCRIPTION: The content should go through the sequence of calls for a valid LMSGetValue(<parameter>) as defined in the application profile. For the purposes of this test, it does not matter which parameter is sent (although this should be one that is supported in the profile, so that a suitable return value is supplied).

- Response = valid return value
  PASS: Content can handle the response; go to test 9.3.2.7
  FAIL: Content reports an error, or closes down for some reason.

- Response = invalid return value / no response
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

UNTESTED: The content does not make any use of LMSGetValue during the test run.

#### 9.3.2.4 Set Test

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.2.1 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (not specified in the XSD)

DESCRIPTION: The content should go through the sequence of calls for a valid LMSSetValue(<parameter>,<value>) as defined in the application profile. For the purposes of this test it does not matter which parameter and value are sent.

- Response = “true”
  PASS:  Content can handle the response; go to test 9.3.2.7
  FAIL: Content reports an error, or closes down for some reason.

- Response = “false”
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

- Response = anything that is not in the specification / no response
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

UNTESTED: The content does not make any use of LMSSetValue.

#### 9.3.2.5 Commit Test

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.2.1 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (not specified in the XSD)

DESCRIPTION: The content should go through the sequence of calls for a valid LMSCommit() as defined in the application profile. For the purposes of this test it does not matter which parameter and value are sent.

- Response = “true”
  PASS: Content can handle the response; go to test 9.3.2.7
  FAIL: Content reports an error, or closes down for some reason.

- Response = “false”
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

- Response = anything that is not in the specification / no response
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content carries on regardless.

UNTESTED: The content does not make any use of LMSCommit.

#### 9.3.2.6 Error Handling Tests (Error found)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1 and 3.3.3 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

**NB:** If this test is being run due to redirection from another method, then only a valid response should be generated; as it is the other method that is under test. If testing content handling off error methods, then all three of the below should be tested.

DESCRIPTION: Following any call, content must make a call to LMSGetLastError() to ensure that the call that was made previously has returned correctly. In these tests, it is assumed that the previous call failed in some way. In this instance, the responses from the server to this call can be either:

- Valid response: Any error code that is specified in the specification for the call made.
  PASS: Go to test 9.3.2.8

FAIL: Test 9.3.2.8 does not occur; The content ignores the response and carries on regardless.

- Invalid response: Any other error code (including 0), or a fictitious error code.
  PASS: The content reports that an invalid error code has been returned. The content should realise that the system is broken and the user MUST be warned of this.
  FAIL: The content ignores the error and carries on regardless, or execution ends abnormally .

- No response
  PASS: The content reports that the connection to the LMS has been lost. The content should realise that the system is broken and the user must be warned of this.
  FAIL: The content ignores the error and carries on regardless, or execution ends abnormally .

### 9.3.2.7 Error Handling Tests (Success found)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1 and 3.3.3 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

**NB:** If this test is being run due to redirection from another method, then only a valid response should be generated; as it is the other method that is under test. If testing content handling off error methods, then all three of the below should be tested.

DESCRIPTION: As in 9.3.2.6, but with a somewhat narrower range of allowed return values.

- Valid response: Error code “0” is returned.
  PASS: The content does nothing, and continues with execution.
  FAIL: The content flags up an error.

- Invalid response: Any other error code, or a fictitious error code.
  PASS: The content reports that an invalid error code has been returned. The content should realise that the system is broken and the user MUST be warned of this.
  FAIL: The content ignores the error and carries on regardless, or execution ends abnormally.

- No response
  PASS: The content reports that the connection to the LMS has been lost. The content should realise that the system is broken and the user must be warned of this.
  FAIL: The content ignores the error and carries on regardless, or execution ends abnormally .

### 9.3.2.8 Error Handling Test (get error string)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1 and 3.3.3 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

DESCRIPTION: Following the return of a non zero error code, content must interrogate the server for an error string. The responses are:

- Valid response: A string of characters is returned.
  PASS: The content acts on the string.
  FAIL: Call is not made.

- No response:
  PASS: The content reports that the connection to the LMS has been lost. The content should realise that the system is broken and the user must be warned of this.
  FAIL: The content ignores the error and carries on regardless, or execution ends abnormally.

#### 9.3.2.9 Bogus calls

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.3.2.1 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (not specified in the XSD)

DESCRIPTION: The content must not attempt to make any bogus API calls to methods that do not exist.

PASS: No bogus API calls are made during the lifetime of the run.

FAIL: Bogus API calls are made.

#### 9.3.2.10 Finish Test

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1, 3.3.2.2 and 3.3.6.1 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not specified in the XSD)

DESCRIPTION: The last call from the content should be to LMSFinish(“”).

PASS: The last call from the content is LMSFinish(“”).

#### 9.3.2.11 FAIL: The last call from the content is not LMSFinish(“”)Finish Test – Responses

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.3.2.1, 3.3.2.2 and 3.3.6.1 / UFI Application Profile

TEST TYPE: Base Specification SCORM Run-time (not XSD)

DESCRIPTION: Given a call from the content to LMSFinish(“”), the content should be able to handle all of the responses below.

- Response = “true”
  PASS: Content can handle the response; content closes down.
  FAIL: Content reports an error, does not close down.

- Response = “false”
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content closes down.

- Response = “anything that is not in the specification” / no response
  PASS: Go to test 9.3.2.6
  FAIL: Test 9.3.2.6 does not occur; content closes down regardless.

### 9.3.3 Content Interaction Tests

These tests ensure that content only passes valid information to and from an LMS. Each of these tests are to ensure that get and set calls pass the correct type of information, and that the responses that come back are returned. Where the previous section is concerned with the sequencing and responses from API calls, this section is concerned with ensuring that the data passed between content and LMS is valid.

#### 9.3.3.1 Read test (valid attribute)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The content attempts to “get” an attribute that is specified as being read only, or read/write in the specification / profile. (As stated the order of calls is not being tested here; these are covered in the previous sections). Each different attribute that the content gets in its lifetime should be tested. The server must always return a valid value for the attribute being requested.

PASS: The content processes the data and continues with execution.

FAIL: The content reports errors, or execution ends abnormally .

#### 9.3.3.2 Read test (invalid attribute)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The content attempts to “get” an attribute that is specified as being write only. This should never occur within a lifetime of content execution.

PASS: The content never attempts to do this during the life of the test.

FAIL: The content attempts to do this.

#### 9.3.3.3 Write test (valid attribute)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The content attempts to “set” an attribute that is specified as being write only, or read/write in the specification / profile. (As stated the order of calls is not being tested here; these are covered in the previous sections). Each different attribute that the content sets in its lifetime should be tested. The server must return true. (The data type is not being tested here; that is covered in later tests).

PASS: The content processes the data and continues with execution.

FAIL: The content reports errors, or execution ends abnormally .

#### 9.3.3.4 Write test (invalid attribute)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The content attempts to “set” an attribute that is specified as being read only. This should never occur within a lifetime of content execution.

PASS: The content never attempts to do this during the life of the test.

FAIL: The content attempts to do this.

#### 9.3.3.5 Array Handling

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The dot notation is correct for the get AND set of an array value. This should be performed in tandem with tests 8.3.5.1 through 8.3.5.4. For clarity the array test is described once here and not four times.

PASS: Attribute set or got is a valid array element – content continues.

FAIL: Attribute set or got is an invalid array element – error codes returned are ignored and content continues.

#### 9.3.3.6 Containers

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: The content should not attempt to get or set a container attribute in the dot notation. This should be performed in tandem with tests 8.3.5.1 through 8.3.5.4. For clarity this test is described once.

PASS: The content never attempts to get or set a container attribute during the life of the test.

FAIL: The content attempts this.

#### 9.3.3.7 GetChildren calls

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: calls to LMSGet(<attribute>._children), are only allowed from certain attributes as noted in the specification and profile. This test should be performed in tandem with tests 8.3.5.1 through 8.3.5.2.

PASS: Calls to the get children are in the correct places; and the content handles the responses.

FAIL: The calls are in the wrong places, error codes are ignored and the content carries on regardless.

#### 9.3.3.8 Error Code validation (expected codes)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.4.4 and 3.3.3

TEST TYPE: Application Profile SCORM Run-time (not specified in the XSD)

DESCRIPTION: When content receives error codes, it should be able to handle the codes that are received. The specification details which error codes are returned for getting / setting a particular attribute. The content must be able to handle all codes that are appropriate for the call made.

PASS: The content can handle any valid error codes that are returned (see test 9.3.3.9 for invalid codes), and continues appropriate execution.

FAIL: The content is unable to handle any codes that are returned so that execution terminates abnormally or content stops responding (hangs).

#### 9.3.3.9 Error code validation (unexpected codes)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Sections 3.4.4 and 3.3.3

TEST TYPE: Application Profile SCORM Run-time (not specified in the XSD)

DESCRIPTION: When content receives error codes that are unexpected for the type of call that has been made, then it should report to the user that the underlying LMS is unstable.

PASS: The content recognises that it has received a bogus error code and handles this appropriately. This response indicates a serious error with the LMS and the user should be informed.

FAIL: The content carries on regardless.

#### 9.3.3.10 Data Types are valid (sent)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: Content sends data to the server that has to conform to the details of the specification / profile for that particular element. Where the profile states that data should be an integer between 1 and 100, for example, then content should send an appropriate value.

PASS: For all calls, an appropriate data value is sent to the server.

FAIL: Content sends inappropriate data.

#### 9.3.3.11 Data types are valid (valid received)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: Content receives data from the server that should conform to the details of the specification / profile for that particular element. For this test, the server returns valid data for the type requested.

PASS: Data received is processed, and execution continues.

FAIL: Data received is processed, and is incorrectly identified as erroneous or execution ends abnormally.

#### 9.3.3.12 Data types are valid (invalid received)

REFERENCED SPECIFICATION: SCORM v1.2 Run-time Environment Section 3.4.4 / UFI Application Profile

TEST TYPE: Application Profile SCORM Run-time (specified in the XSD)

DESCRIPTION: Content receives data from the server that is of the wrong type or out of range from the profile for the particular element.

PASS: Data received is processed, and the content realises that there is a serious error with the LMS, and the user is informed.

FAIL: Data is received, and the content continues execution regardless.

### 9.3.4 Content Package Tests

The following tests are applied to content packages.

#### 9.3.4.1 Decompression test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 2.1 (package interchange file)

TEST TYPE: Base Specification IMS Content Packaging (not specified in the XSD)

DESCRIPTION: The test system will attempt to decompress the content package. The resulting files will then be dumped onto the file system.

PASS: The test system is able to decompress the PIF.

FAIL: The test system is unable to decompress the PIF.

UNTESTED: The content is packaged on a CD or other media and not as a single file.

#### 9.3.4.2 Manifest Presence Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 2.2

TEST TYPE: Base Specification IMS Content Packaging (not specified in the XSD)

DECRIPTION: The imsmanifest.XML file must exist at the root level of the content package and must be written in lowercase letters only.

PASS: The imsmanifest.XML file is located at the root level of the package and is written in lowercase letters.

FAIL: The imsmanifest.XML file is not located at the root level of the package or not written completly in lowercase letters.

#### 9.3.4.3 Manifest schema Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 2.2

TEST TYPE: Base Specification IMS Content Packaging (not specified in the XSD)

DESCRIPTION: The Content Packaging specification states that all supporting schemas and other such files that are required by the manifest should exist at the root level of the package. Thus, any dependent schemas must first be determined from the manifest, and then their presence confirmed at the root level of the package.

PASS: All dependent schemas are located at the root level.

FAIL: Any dependent schemas are not located at the root level.

#### 9.3.4.4 Manifest validation Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 4 / UFI Application Profile for Content Packaging

TEST TYPE: Application Profile IMS Content Packaging (specified in the XSD)

DESCRIPTION: the manifest file located in 8.3.2.2 must now be tested as an XML file as per the descriptions in 8.3.3.

PASS: See section 8.3.3

FAIL: See section 8.3.3.

#### 9.3.4.5 Manifest Resource Presence Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 2.1 (resource)

TEST TYPE: Base Specification IMS Content Packaging (not specified in the XSD)

DESCRIPTION: The manifest should be parsed and the presence of all referenced resources should be verified.

PASS: All referenced resources can be located (that is that they are present (as shown by the <file> element) in the content package, or if remote resources, then the files are found over the internet).

FAIL: Any referenced resource cannot be located.

#### 9.3.4.6 Manifest Completeness Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 2.1 (Key elements)

TEST TYPE: Base Specification IMS Content Packaging (not specified in the XSD)

DESCRIPTION: The manifest should reference everything that is within the content package. A list of all the resources that are present in the content package (except the manifest itself) should be compiled and then this should be checked against the manifest.

PASS: No files are present in the package that are not listed in the manifest.

FAIL: Files exist in the package that are not listed in the manifest.

#### 9.3.4.7 Organisation Default Test

TEST TYPE: Base Specification IMS Content Packaging (specified in the XSD)

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 4 / UFI Application Profile

DESCRIPTION: The default organisation contains an IDREF which must only point to the ID of an organisation.

PASS: The default IDREF for organisations points to the ID of an organisation.

FAIL: The default IDREF for organisations points to a non existent ID, or to an ID that does not belong to an organisation element.

#### 9.3.4.8 Item IdentifierRef to Resource Identifier Test

REFERENCED SPECIFICATION: IMS Content Packaging Information Model version 1.1.1 Section 4 / UFI Application Profile

TEST TYPE: Base Specification IMS Content Packaging (specified in the XSD)

DESCRIPTION: The item element contains an IdentifierRef which must point to the identifier of a resource in the manifest.[Only in the UFI application profile – in the spec, this is also allowed to point to other manifests]

PASS: For each item in the manifest, the IdentifierRef points to the identifier of a resource in the manifest.

FAIL: For any item in the manifest, the IdentifierRef points to a non existent ID, or to an ID that does not belong to a resource element.

### 9.3.5 XML File Tests

The following tests are applied to XML files; be they manifests or metadata files.

#### 9.3.5.1 Well Formed

REFERENCED SPECIFICATION: Extensible Markup Language (XML) 1.0, Section 2.1

TEST TYPE: Base Specification [IMS MD / LOM MD / IMS CP] XSD)

DESCRIPTION: The XML within the file must be well formed.

PASS: The XML is well formed.

FAIL: The XML is not well formed.

#### 9.3.5.2 Internal Validation

REFERENCED SPECIFICATION: Extensible Markup Language (XML) 1.0, Section 2.1

TEST TYPE: Base Specification IMS MD / LOM MD / IMS CP (XSD)

DESCRIPTION: If the XML refers to DTD documents or XML schemas, then the XML must be valid against them.

PASS: The XML validates without error against the DTD or Schema.

FAIL: The XML fails to validate.

UNTESTED: If the XML does not refer to any other documents, then this test does not need to be performed.

#### 9.3.5.3 Non-conditional Restraints Test

REFERENCED SPECIFICATION: D4 – Application Profiles.

TEST TYPE: Application Profile IMS MD/ LOM MD/ IMS CP (XSD)

DESCRIPTION: The XML file under test must conform to all non-conditional restraints expressed within the application profile. [NOTE - this is the test of the resultant schema]

PASS: The XML document successfully validates against all the non-conditional rules expressed in the application profile.

FAIL: The XML document fails to validate against at least one of the non-conditional rules expressed in the application profile.

#### 9.3.5.4 Conditional Restraints Test

REFERENCED SPECIFICATION: D4 – Application Profiles.

TEST TYPE: Application Profile IMS MD/ LOM MD/ IMS CP (XSD)

DESCRIPTION: The XML file under test must conform to all conditional restraints expressed within the application profile. This means that the XML files under test successfully validates against each conditional rule.  [NOTE  - this is the test of the conditional file]

PASS: The XML document successfully validates against all conditional rules.

FAIL: The XML document does not validate against at least one of the conditional rules.

# 10 CONFORMANCE CLAIMS

## 10.1 Designations of conformance.

A test can have exactly one of the following states:

- Pass
- Fail
- Untested

If a product fails any of the tests then it is not conformant.

If a product passes all of the tests that it is tested against, then it is conformant. At this stage, there are no levels of conformance other than a binary pass or fail.

# 11 APPENDIX – UFI TESTS NOT RELATED TO DOCUMENTS LISTED ABOVE

## 11.1 Software

REFERENCED SPECIFICATION: UFI Cad document.

TEST TYPE: Other

- Content must be able to be executed from a web server. As a result the content must be able to be executed from all web browsers.
- Content packages must be able to be unzipped using WinZip and GZIP.
- XML files should be able to be viewed in a HTML 4.01 compliant web browser without errors being produced.
- For extended content like Shockwave or Flash the corresponding player or browser plug-in must be available for free on the internet.

## 11.2 Network

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

- Content must be able to function at a configurable network speed (default 35kbs).
- A “page” or “screen” of content must load within a configurable time (default 10 seconds). If the load time is more than a configurable time (default 5 seconds), a suitable animation must be displayed to advise end users of the delay.
- Network is not a consideration for XML files or content packages.
- HTTP-based communication must be able to function even if the client and the server have no direct point-to-point communication, e.g.: If they must communicate through a HTTP proxy or the client uses an unroutable internal IP addresses with NAT-based (Network Address Translation) connection.

## 11.3 Physical environment

REFERENCED SPECIFICATION: UFI cad document

TEST TYPE: Other

- Content must fit into a configurable screen size (default 724x494) and be fully operational at a configurable screen resolution (default is 800x600).
- Content must not be larger than a configurable (default 1gb) size.

## 11.4 Consolidated Test List

### 11.4.1 Environmental Tests

Tests that are confined to the hardware / software / networking environment within which the content is to be run.

For the platform tests, the assumption is that the test system is deployed on the architecture presented in the test, and content is uploaded to the system.

The envisaged test system is ultimately a web based system where the users will interact using a web browser. Users of the system are expected to use Windows or Apple Macintosh workstations, and will use a variety of web browsers. The tests in this section enumerate client browsers and platforms. It is important to note, however, that as browser technology continues to develop, so the system should be usable from new browsers.

#### 11.4.1.1 Platform Test (Intel)

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when hosted on an Intel / Windows / Apple platform. {Note that Intel / Solaris x86 is not considered}.

PASS: Content can execute

FAIL: Content cannot execute

UNTESTED: Content documentation specifies that the content is not supported on this server platform.

#### 11.4.1.2 Platform Test (Sun)

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when hosted on a Sun / Solaris platform.

PASS: Content can execute

FAIL: Content cannot execute

UNTESTED: Content documentation specifies that the content is not supported on this server platform.

#### 11.4.1.3 Browser test: Windows / Opera / Mozilla

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when used from a windows / opera / Mozilla web browser

PASS: Content can execute.

FAIL: Content cannot execute.

UNTESTED: Content documentation specifies that the content is not supported on this client platform.

#### 11.4.1.4 Browser test: Windows / Internet Explorer v6

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when used from a windows / internet explorer v6 web browser.

PASS: Content can execute.

FAIL: Content cannot execute.

UNTESTED: Content documentation specifies that the content is not supported on this client platform.

#### 11.4.1.5 Browser test: Mac / Internet Explorer v5

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when used from a Mac / internet explorer v5 web browser.

PASS: Content can execute.

FAIL: Content cannot execute.

UNTESTED: Content documentation specifies that the content is not supported on this client platform.

#### 11.4.1.6 Browser test: Mac / Safari

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when used from a Mac / Safari web browser.

PASS: Content can execute.

FAIL: Content cannot execute.

UNTESTED: Content documentation specifies that the content is not supported on this client platform.

#### 11.4.1.7 Browser Test: Linux / Sun Java Desktop Browser

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: Content must be functional when used from a Linux / java desktop web browser. (for more information see http://wwws.sun.com/software/javadesktopsystem/details.html).

PASS: Content can execute.

FAIL: Content cannot execute.

UNTESTED: Content documentation specifies that the content is not supported on this client platform.

#### 11.4.1.8 Network Speed Test

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

DESCRIPTION: The content must be able to function at a configurable network speed. (default 35kbs).

PASS: Content is usable at this network speed.

FAIL: Content is unusable at this network speed.

### 11.4.2 General Content Tests

These are tests of a general nature that must be applied to all content under test.

#### 11.4.2.1 Content Resolution Test

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

DESCRIPTION: Content must run at a configurable screen resolution (default 800 by 600).

PASS: Content runs at the screen resolution, and none of the elements of the content are missing or otherwise not displayable.

FAIL: The content fails to run at this screen resolution, or elements of the content do not display correctly.

#### 11.4.2.2 Content Screen Size Test

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

DESCRIPTION: Content must fit into a configurable screen area (default 724x494) with all of the content able to display correctly.

PASS: Content displayed correctly at this screen size.

FAIL: Content does not display correctly at this screen size.

#### 11.4.2.3 Content page load test

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

DESCRIPTION: Content must load each ‘page’ within 10 seconds of the page being opened.

PASS: Content loads each page within 10 seconds

FAIL: Content fails to load each page within 10 seconds.

#### 11.4.2.4 Content Animation test

REFERENCED SPECIFICATION: UFI Cad document

TEST TYPE: Other

DESCRIPTION: If content is taking longer than 5 seconds to load a page, then an animation or message must be displayed to the user to advise them of the delay.

PASS: Animation / message displayed on a slow load

FAIL: Animation / message not displayed on a slow load.

#### 11.4.2.5 Graceful degradation ‘test’

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: At no point during execution should content crash, hang, become unresponsive or otherwise end abnormally. Content should never require a system process to be killed to terminate its execution. This behaviour occurring at any time during testing will fail that test.

PASS: Content never crashes etc during testing.

FAIL: Content exhibits the behaviour above during testing.

#### 11.4.2.6 Content spelling test

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: The spelling of all words displayed in the content should be verified against an external dictionary to ensure that all words are spelt correctly as defined against a reference language for the Application Profile. Users should not see any words that are incorrectly spelt.

PASS: All words in the content are correctly spelt.

FAIL: Any words

#### 11.4.2.7 Content grammar test

REFERENCED SPECIFICATION: UFI Internal

TEST TYPE: Other

DESCRIPTION: The grammar of all textual elements of the content should be checked to ensure that it is valid and well formed as defined against a reference language for the Application Profile.

PASS: Grammar is correct within the content.

FAIL: Grammar contains errors.

### 11.4.3 Content Package Tests

The following tests are applied to content packages.

#### 11.4.3.1 Content Size

REFERENCED SPECIFICATION: UFI Cad Document

TEST TYPE: Other

DESCRIPTION: The Content should not be greater than a configurable (default 1gb) size

PASS: The content is less than the size specified.

FAIL: The content is not less than the size specified.