



## **Marine Corps Distance Learning Network (MarineNet)**

---

### **Courseware Development**

### **Technical Manual**

Version 2.1  
FINAL

February 2004

*Note: This document includes the requirements to implement the SCORM specification and identifies specific limitations on the MarineNet system in respect to SCORM. This document does not replace the requirements documented for vendors currently developing toward the AICC requirement documented in a previous version of this document.*

College of Continuing Education (CCE)  
Training & Education Command (TECOM)  
United States Marine Corps

## **ABSTRACT**

These guidelines outline the technical specifications for content being delivered on MarineNet. Marine Corps College of Continuing Education (CCE) is committed to promoting reusability, durability, accessibility, interoperability, maintainability, and portability of distance learning content. However, these guidelines primarily address content interoperability issues.

The technical guidelines outlined in this document represent a combination of specifications for the various systems and technologies implemented by MarineNet. Requirements identified in this document reference the target deployment environments. Courseware vendors developing courseware for hosting on MarineNet should consult their Statement of Work (SOW) to determine the target deployment environments. Courseware SOWs supersede requirements defined in this document.

Updates to this document will be made periodically to provide further guidance to developers.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>1.1</b>	<b>AICC and SCORM Standards .....</b>	<b>1</b>
<b>1.2</b>	<b>Web Accessibility Guidelines .....</b>	<b>2</b>
<b>2</b>	<b>SYSTEM SPECIFICATIONS .....</b>	<b>3</b>
<b>2.1</b>	<b>Hardware .....</b>	<b>3</b>
<b>2.2</b>	<b>Operating Systems .....</b>	<b>4</b>
<b>2.3</b>	<b>Browsers.....</b>	<b>4</b>
<b>2.4</b>	<b>Plug-ins.....</b>	<b>5</b>
<b>2.5</b>	<b>MarineNet Distributed Architecture.....</b>	<b>5</b>
2.5.1	MarineNet Intranet Environment .....	6
2.5.2	MarineNet Internet Environment .....	6
2.5.3	Use of Server Variables in a Distributed Environment .....	7
<b>2.6</b>	<b>Network Connectivity: High-Bandwidth &amp; Low-Bandwidth.....</b>	<b>8</b>
<b>3</b>	<b>MULTIMEDIA .....</b>	<b>10</b>
<b>3.1</b>	<b>Static Image Formats.....</b>	<b>10</b>
3.1.1	Image File Formats .....	10
<b>3.2</b>	<b>Macromedia Flash and Shockwave Content .....</b>	<b>10</b>
<b>3.3</b>	<b>Audio and Video Formats .....</b>	<b>11</b>
3.3.1	Streaming Media Codecs .....	11
3.3.2	Streaming Media Encoding Rates.....	11
3.3.3	Streaming Media References .....	12
3.3.4	Embedding Media Files .....	13
3.3.5	Active Content in Internet Explorer.....	13
<b>4</b>	<b>ACCEPTABLE RESPONSE TIMES .....</b>	<b>15</b>
<b>4.1</b>	<b>Multimedia Response Times .....</b>	<b>15</b>
<b>4.2</b>	<b>Overall Page Response Times .....</b>	<b>15</b>
<b>5</b>	<b>CONTENT PROGRAMMING .....</b>	<b>16</b>
<b>5.1</b>	<b>Static Content .....</b>	<b>16</b>
<b>5.2</b>	<b>Dynamic Content .....</b>	<b>16</b>
<b>5.3</b>	<b>Screen Sizes and Programming .....</b>	<b>16</b>
<b>5.4</b>	<b>Content Packaging .....</b>	<b>16</b>
5.4.1	<i>file.CRS</i> .....	<b>Error! Bookmark not defined.</b>
5.4.2	<i>file.AU</i> .....	<b>Error! Bookmark not defined.</b>
5.4.3	<i>file.DES</i> .....	<b>Error! Bookmark not defined.</b>
5.4.4	<i>file.CST</i> .....	<b>Error! Bookmark not defined.</b>
<b>5.5</b>	<b>Directory Structure.....</b>	<b>17</b>
<b>5.6</b>	<b>File Naming Conventions .....</b>	<b>17</b>
<b>5.7</b>	<b>Initial Launch File.....</b>	<b>17</b>
<b>5.8</b>	<b>Acceptable Software Programming Tools and Aids.....</b>	<b>17</b>
<b>5.9</b>	<b>Mobile Code.....</b>	<b>18</b>
5.9.1	Mobile Code Category Descriptions.....	18
5.9.2	Mobile Code Designations.....	19

**5.10 Code Reuse and Open Source..... 19**

**6 INTERACTIONS WITH THE MARINETET LEARNING MANAGEMENT SYSTEM (LMS) ..... 21**

**6.1 Relationship between Courseware and the MarineNet Course Catalog ..... 21**

**6.2 Courseware Launch Window Sizing..... 21**

**6.3 Interfacing with the SCORM-conformant MarineNet LMS ..... 22**

6.3.1 SCORM Limitations in the MarineNet LMS..... 22

6.3.2 Additional SCORM Development Guidelines..... 25

**7 TESTS AND SURVEYS..... 29**

**7.1 Pre-assessments, In-line Quizzes, Practice Tests, and Surveys ..... 29**

**7.2 MarineNet LMS End-of-Course (EOC) Tests..... 29**

7.2.1 Learning Objectives Worksheet..... 29

7.2.2 Test Questions Worksheet ..... 31

7.2.3 Valid Responses Worksheet ..... 32

7.2.4 Administrative Worksheet ..... 34

**8 FUTURE TRENDS..... 36**

**8.1 Client Software Baselines: Browsers and Plug-ins ..... 36**

**8.2 Internet-based Content: LMS Tracking and Performance ..... 36**

**8.3 AICC, SCORM, and other LMS Interface Standards ..... 36**

**8.4 Legacy MarineNet Content..... 37**

## 1 INTRODUCTION

MarineNet is a complex system that utilizes and incorporates the following:

- Centralized Learning Management System (LMS) services using a customized THINQ product,
- Remotely located commercial content distribution and delivery services,
- Marine Corps Enterprise Network infrastructure, and
- Interactive instructional content and static web-based information resources.

The MarineNet Program Office is co-managed by the CCE and MARCORSYSCOM, both located at Quantico, VA. MarineNet is supported through a group of support contractors and a partnered government organization.

There are many technical and instructional issues to be considered when developing content for delivery on MarineNet. The evolution of the Sharable Content Object Reference Model (SCORM) and Aviation Industry Computer-Based Training Committee (AICC) guidelines coupled with the rigid architecture provided by the Marine Corps Enterprise Network (MCEN) amplifies the critical need for specific guidelines for content development, packaging, distribution, hosting, and progress tracking.

Developers should have a basic understanding of the underlying AICC and SCORM development guidelines as well as general web technologies and accessibility issues governed by the Department of Defense (DoD) and the USMC. This document is not intended to replace guidance provided by other specification bodies. Instead, it is intended to clarify which specific requirements in other documents the developers are expected to follow that are relevant to MarineNet. It will also identify any additional requirements imposed by the MarineNet architecture.

### 1.1 AICC and SCORM Standards

AICC Guidelines and Recommendations (AGRs) are widely accepted in industry for developing Web-based training. SCORM is also becoming a widely accepted industry standard.

Currently, the AICC has two protocols for communicating with an LMS - AICC HyperText Transfer Protocol (HTTP) and AICC Application Program Interface (API). The SCORM has one protocol for communicating with an LMS - SCORM API. The AICC API and the SCORM API protocols are identical, which means that content using the AICC API is compatible with the SCORM API. However, the AICC HTTP protocol is incompatible with the SCORM API.

The MarineNet system supports both the AICC HTTP and SCORM protocols. However, there are some restrictions involved with API protocols operating in a distributed environment (see Section 2.5.3).

## 1.2 Web Accessibility Guidelines

The Marine Corps is dedicated to making all distance learning content accessible to each person who uses the system. Section 508 of the Rehabilitation Act of 1996 requires all Federal agency electronic and information technology to be accessible to people with disabilities, including employees and members of the public.

Developers must ensure that content is Web-accessible to persons with disabilities. Some of the requirements identified in this document are intended to ensure that content delivery meets Web-accessibility guidelines. All content developed for the Marine Corps shall be Section 508-compliant. The degree of compliance shall vary based on the type of development and the target audience. Developers shall consult their Government Program Manager (PM) to identify the required degree of compliance. As a baseline, all content must meet the Conformance Level "A" requirements identified in the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines version 1.0. Developers are encouraged to read the W3C Web Accessibility Guidelines at the W3C Website, <http://www.w3c.org/WAI/resources>.

## 2 SYSTEM SPECIFICATIONS

The MarineNet environment includes a wide range of target audience workstations. Workstations on Marine Corps installations are typically managed workstations with strict hardware and software configurations. The Marine Corps Common Hardware Suite (MCHS) requires all Marine Corps workstations meet certain minimum configurations. Base G-6 offices govern what software is permissible to be loaded on MCHS workstations. Bases undergoing conversion to the Navy Marine Corps Intranet (NMCI) management operate NMCI workstations with strict hardware and software configurations. Lastly, Learning Resource Center (LRC) workstations use MCHS hardware with software loads managed by the CCE.

The applicable software baselines are available at:

- MarineNet Software Baseline:  
[http://www.tecom.usmc.mil/cce/references/technical/ref\\_software\\_baseline.asp](http://www.tecom.usmc.mil/cce/references/technical/ref_software_baseline.asp)
- NMCI Gold Load:  
[http://www.nmci-eds.com/downloads/Gold\\_disk\\_contents.pdf](http://www.nmci-eds.com/downloads/Gold_disk_contents.pdf)
- MCHS:  
<http://mchs.marcorsyscom.usmc.mil/>

While it can be expected that all workstations on military installations will have strict software loads, Internet users may operate varying configurations. This section will establish a baseline development platform. Although the established baseline will not cover all possible configurations, the intent is to simplify development in order to reduce development costs and decrease the time to market for newer courses. The minimum hardware and software requirements to access MarineNet are delineated in the MarineNet Hardware and Software Baseline.

### 2.1 Hardware

Developers shall ensure their content runs as intended on all systems meeting the following minimum hardware specifications. Unless specified by the government PM, the Internet version of hardware standards should be used.

**Internet / Personal Workstation Standards:**

- IBM compatible PC or equivalent
- Microprocessor: Pentium 450 MHz
- Hard Drive: 10 GB
- RAM: 64 MB
- Sound Card: 16 bit
- Speakers
- CD-ROM Drive: 12x
- Monitor Resolution: 1024x768
- Color Depth: 24 bit
- Fonts: Small
- Network Connection: 10 Mbps Network Interface Card (NIC) (Replaced by 56Kbps Modem for low-bandwidth dial-up users)

**Intranet / Marine Corps / NMCI Workstation Standards:**

- IBM compatible PC or equivalent
- Microprocessor: Pentium III 550MHz
- Hard Drive: 10 GB
- RAM: 128MB RAM
- Sound Card: 16 bit
- Headphones
- CD-ROM Drive: 40X
- Monitor Resolution: 1024x768
- Color Depth: 24 bit
- Fonts: Small
- Network Connection: 10 Mbps Network Interface Card (NIC)

**2.2 Operating Systems**

The Marine Corps workstations are currently running Microsoft Windows 2000. All content developed to operate on MarineNet must be fully compatible with this operating system. Courseware developed to be accessible to Internet users (low-bandwidth or high-bandwidth) shall be compatible on Microsoft Windows 98, ME, 2000, and XP operating systems. Further requirements may be identified by the Government PM.

**2.3 Browsers**

Developers shall design learning content optimized for Internet Explorer 6.0. MarineNet is not fully compatible with Netscape. Further requirements may be identified by the Government PM.

## 2.4 Plug-ins

All content must be compatible with all of the potential software configurations defined by the software baselines outlined in section 2. Furthermore, to simplify configuration management and reduce the cost of lifecycle maintenance courseware developers shall be further limited to use only certain technologies, where practical.

All MarineNet developed content shall be programmed to use the technologies based on these plug-ins:

- Adobe Acrobat Reader version 5.0
- Macromedia Shockwave version 8.5
- Macromedia Flash Player 5
- Microsoft Windows Media Player (WMP) 7.01.00.3055
- Microsoft Virtual Machine & Sun Java Plug-in

As previously stated the above technologies shall be used whenever possible. The software baselines may permit the use of other technologies such as Real Player, QuickTime, etc., however, these technologies shall only be used if the technologies listed in this section cannot meet the instructional intent of the content. The use of any other plug-in not listed above must be approved by the Government PM.



*Important: Internet users may not have the plug-ins preloaded on their workstations. Therefore, all courseware shall provide the user with a list of the required plug-ins for that course. This list of required plug-ins shall be provided in the courseware at a location defined by the MarineNet Standard Interface. The courseware shall also link to the MarineNet Portal Site allowing the user to download the plug-ins.*



*Important: Any content object developed in Java, therefore requiring a Java Run-time Environment, shall be developed to be compatible to both the Microsoft VM as well as the Sun Java Run-time Environment. These content objects shall be developed to be compatible to all versions of the Run-time Environments unless the limitation is approved by the Government PM.*

## 2.5 MarineNet Distributed Architecture

The MarineNet distributed architecture refers to the distributed nature of the content. Clients operate in two environments: Intranet and Internet. The Intranet environment includes all users on military installations that host a local content server. The Internet environment includes all other locations.

### 2.5.1 MarineNet Intranet Environment

While the MarineNet LMS operates as a centralized capability located on a Military installation, all content resides on content servers located near the users. The primary reason for this architecture is two fold. First, content files tend to be large in size and consume significant bandwidth. Secondly, Marine Corps' installations operate border firewalls that "strip out" mobile code and object inclusions (such as streaming media controls) from HTML/ASP content.

Based on the configuration of firewall boundaries and location of distributed content servers, several conclusions can be made:

- Content cannot be centrally located unless it contains only standard HTML code without JavaScript, Java, ActiveX, object includes, or streaming media.
- Content must be specially formatted to permit the MarineNet content management system to package and distribute the content to the remote content servers.
- The use of other service, agency, organization, or developer's content must be carefully tested for compatibility with LMS interfaces, content distribution, content hosting, client compatibility, and security.



*Important: Marine Corps installations that do not host a local content server cannot access the Internet capability for content that contains mobile code due to firewalls. All Marine Corps firewalls block content containing JavaScript, Java, ActiveX, object includes, or streaming media.*

### 2.5.2 MarineNet Internet Environment

It can be assumed that there are no firewalls between the Internet users and the Internet content Server. Internet Service Providers (ISPs), other local and metropolitan area networks, and individual users may operate firewalls. These firewalls are outside the control of the MarineNet system, however, they may restrict an individual user's access to content. The MarineNet system will provide information to help users determine the source of their problem and help rectify the problem. This information will be located on the MarineNet Portal Site. Users operating workstations in locations that operate firewalls that are controlled by policies outside the control of the individual users may be restricted from accessing some MarineNet content. This situation cannot be avoided and the user may experience sporadic and/or critical problems accessing certain content elements. They will not receive on-screen instructions of why the problems are occurring.

Internet users can be expected to operate workstations without the required plug-ins and software. Therefore, the MarineNet system will be responsible for providing instructions on how to obtain and load such software and plug-ins. This information will be located on the MarineNet Portal Site. It is assumed that all Internet users will have the ability to load and configure such software and plug-ins.

### 2.5.3 Use of Server Variables in a Distributed Environment

The use of server variables in a distributed environment presents several potential problems. This section outlines some of the potential problems and restrictions for using server variables in the MarineNet environment.

#### 2.5.3.1 SCORM and AICC Application Programming Interfaces (API)

The SCORM and AICC API protocols are built on the interaction with server variables located within the LMS. Courseware interacting with these protocols must have access to these variables in order to interface with the LMS. In a distributed environment, this is restricted since server variables cannot be accessed outside of the Internet Server's process namespace. This means that all files accessing server variables must be located on the same virtual server or within the same virtual directory space. The Web server providing the virtual links may be on a different physical server; however, all of the code must be associated via a virtual directory on the web server. In the MarineNet distributed environment, courseware is located on remote physical servers that do not have virtual directory linkage to the LMS web server and therefore cannot share server variables.

Thus, all interfaces between the courseware and the LMS Web server must rely on stateless or session-less protocols. The AICC HTTP-based AICC CMI Protocol (HACP) provides HTTP functions for courseware to interface with the LMS. These functions require the courseware to maintain information to identify the student's session. This information is relayed to the HTTP functions during each access to the LMS to store or receive information.

MarineNet provides minimal compliance to the SCORM specification through the use of a SCORM Relayer that remotes the SCORM API Adapter, typically provided by the LMS application, to the content servers. The SCORM Relayer is responsible for translating SCORM calls from the SCORM-conformant content to the MarineNet LMS via the AICC HACP protocol.

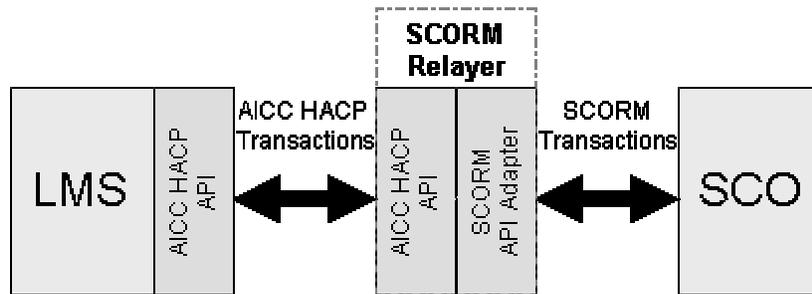


Figure 2-1. SCORM Relayer Interactions Diagram

### 2.5.3.2 *Courseware Web Server Variables and Data Storage*

Courseware may rely on the virtual and physical storage space provided by the courseware Web servers; however the extent of use is restricted. The courseware Web servers operate Microsoft Internet Information Server (IIS) and allow for the use of Active Server Pages (ASP). The application space associated with the ASP configuration allows for courseware content to store server-side variables and run server-side code to communicate with the LMS Web server in a session-based environment. The session-based environment is not “session-based” on the LMS Web server since the session only exists for the duration of the single transmission. However, it does allow for the passing of return codes to the courseware server for function returns and error trapping.

To reduce the processing load, the use of ASP on the content servers is restricted to the basic interfacing with the LMS. Also, the use of server-side variables is restricted to the storage of essential data identifying the student’s session and progress within the course.

### 2.5.3.3 *Server-side and Client-side Web Cookies*

Courseware may not use server-side or client-side web cookies. Web cookies do not function in the MarineNet environment because the content Web servers and the LMS Web server do not always operate in the same domain. The domain scope that allows for the sharing and/or access of cookies is bounded by the Domain Name Service (DNS). Servers within the same tier-two DNS scope (ex. usmc.mil) may access each others’ cookies, however, the MarineNet web servers (LMS and content servers) operate under various DNS names. Furthermore, many content servers are not associated with a DNS name at all.

## 2.6 **Network Connectivity: High-Bandwidth & Low-Bandwidth**

The MarineNet system consists of both Intranet and Internet environments. Intranet environments include Marine Corps Camps, Bases, and Stations as well as other fixed sites that host local content servers. Intranet sites possess high-bandwidth local area networks (LANs) typically providing 400Kbps to 100Mbps connectivity between the local content servers and the user workstations.

MarineNet users at home and at locations that do not host a local content server may not have high-bandwidth connections. The population of MarineNet users that have broadband connections at home is expected to be small, but rapidly increasing. Low-bandwidth users can be expected to possess no more than a 56Kbps modem connection with a typical throughput of approximately 40Kbps or less.

MarineNet courseware, designed for both high-bandwidth and low-bandwidth, shall be programmed to allow the user to determine which version they would like to access.

Users shall be provided the choice of:

1. High-Bandwidth Version. Optimized for a 400Kbps connection between the user and the content server.
2. Low-Bandwidth Version. Optimized for a 40Kbps connection between the user and the content server.

The high- and low-bandwidth versions of the course shall be designed as one LMS course and will exist as one course catalog entry. Users enrolling in the course will have the option of accessing either version at the beginning of each launch session. In addition, any bookmarks stored during the previous session shall be transferable to the other version (high- or low-bandwidth).

Since there are Internet users that have similar high-bandwidth connections as Intranet users, the remainder of this document will use the terms high-bandwidth and low-bandwidth to refer to the two different types of connection speeds. Courseware expected to be accessible to both user types shall comply with all requirements, while courseware developed to only be accessible to one type shall only comply with those, unless otherwise indicated. A low-bandwidth version will not meet the requirements of a high-bandwidth version.



*Important: All courseware shall be developed for the high-bandwidth version unless otherwise directed by the Government PM.*

### 3 MULTIMEDIA

This section describes the requirements for using multimedia files in MarineNet content.



*Important: All multimedia and other courseware files and directories shall not use spaces or any other special characters in file names. Underscores (\_) are permitted.*

#### 3.1 Static Image Formats

Images are the most prevalent form of multimedia used in web-based learning content. Each image file contained in the learning content must adhere to specific requirements designed to meet Web Accessibility guidelines.

##### 3.1.1 Image File Formats

The following list delineates acceptable image file formats that developers can include in their content:

- GIF
- JPG

Courseware developers shall choose the appropriate image file format to best suit the content displayed while also maximizing quality and minimizing file size. All images shall use 72 dots per inch unless the content is specifically designed to be printed.

#### 3.2 Macromedia Flash and Shockwave Content

Macromedia Flash and Shockwave may be used to develop animations and interactive content segments. The maximum size of a Flash or Shockwave file is limited to prevent excessive download times. The table below lists the average and maximum file sizes for high- and low-bandwidth versions.

**File Sizes for Flash and Shockwave Content**

	<b>Average File Size</b>	<b>Maximum File Size</b>
<b>Low-Bandwidth Version</b>	~15 KB	40 KB
<b>High-Bandwidth Version</b>	~50 KB	400 KB

### 3.3 Audio and Video Formats

The following list delineates acceptable audio and video formats that developers can include in their content:

- WMA
- WMV

The table below identifies when audio and video files should be streamed. Section 3.3.2 identifies the streaming rates for streamed media.

#### When to Stream Audio and Video

Type of Multimedia	Low-Bandwidth Version	High-Bandwidth Version
Video	Video Not Permitted	< 400 KB <b>Not</b> Streamed ≥ 400 KB Streamed at Static Rate
Audio	< 40 KB <b>Not</b> Streamed ≥ 40KB Streamed at Static Rate	< 400 KB <b>Not</b> Streamed ≥ 400 KB Streamed at Static Rate

Video (streaming or non-streaming) is not permitted in low-bandwidth versions of courseware. The throughput of a low-bandwidth connection is not conducive to download video files without extensive degradation in quality, thus decreasing the overall learning experience for the user.

#### 3.3.1 Streaming Media Codecs

Windows Media Player 7.01 is the standard player for all streaming media. Streaming video files shall be encoded using the Windows Media Video v7 codec and streaming audio files shall be encoded using Microsoft Windows Media Audio v8 codec.

#### 3.3.2 Streaming Media Encoding Rates

Streaming Media shall be encoded at the following encoding rates.

#### High-Bandwidth Streaming Media Encoding Rates

Type of Media	Rate (Kbps)
Audio	64
Video	200

#### Low-Bandwidth Streaming Media Encoding Rates

Type of Media	Rate (Kbps)
Audio	16
Video	Not Permitted

No more than five seconds of buffering is permitted for streaming content.

### 3.3.3 Streaming Media References

MarineNet requires that courseware developers use an absolute addressing scheme to reference the streaming media files from within the courseware.

The courseware will reference an associated meta-file (.ASX format) for each video or audio file that is encoded in Windows Media format (.WMA) or (.WMV). The current version of MarineNet does not support (.WMX format) metafiles. Audio files may not use the .WAV format.

Example of MediaFileName.ASX:

```
<asx version = "3.0">
  <title>MediaFileName</title>
  <entry>
    <ref href =
mms://www.marinenet.usmc.mil/courses/coursename/module1/lesson3/topic2/media/m
ediafilename.wmv"/>
    <title>MediaFileName</title>
    <abstract>Description of content</abstract>
    <copyright>Department of Defense</copyright>
    <author>United States Marine Corps</author>
  </entry>
</asx>
```



*Important: Courseware developers must use absolute address references for the mms:// protocol when streaming the media. References to non-streaming media must use relative addressing without the identification of the mms: protocol. Courseware references (within the HTML) to the ASX files must be relative.*

The .ASX files will be automatically modified to insert the appropriate content server name in place of **www.marinenet.usmc.mil** in the *href* path during pre-positioning of the content.



*Important: ASX files are still required for non-streaming media, however relative addressing must be used in the ASX files when referencing non-streaming media.*

### 3.3.4 Embedding Media Files

The existing MarineNet courses make extensive use of embedded audio and video. Embedding is accomplished by using the OBJECT tag for the Windows Media Player ActiveX control within the course's code. This OBJECT tag will play the multimedia file directly in the web browser without opening the associated media player. The advantage to embedding the media player is that a developer can control the functionality of the player, its appearance, and also manipulate it through web scripts. A CLASSID is an element of the OBJECT tag used to reference a specific version of the WMP. The syntax is shown in the table below:

WMP Version	Sample OBJECT Tag with corresponding CLASSID
7.0	<pre>&lt;OBJECT ID="MediaPlayer" WIDTH=240 HEIGHT=180 CLASSID="CLSID: 6BF52A52-394A-11D3-B153-00C04F79FAA6" STANDBY="Loading Windows Media Player components..." TYPE="application/x-oleobject"&gt; &lt;PARAM NAME="URL" VALUE="media/mediafilename.asx"&gt; &lt;/OBJECT&gt;</pre>

**!** *Important: Be sure to use the correct CLASSID for the target WMP since there are different CLASSIDs for the WMP versions. The CLASSID for WMP 7.0 is listed in the above example.*

When embedding multimedia, PARAM tags are used within the OBJECT tag to reference the .ASX file. The .ASX file will then reference the streaming file using the format defined on the previous page. A sample PARAM tag is shown above in the OBJECT Tag. PARAM tags can also be used to control the functionality and appearance of the multimedia when it is embedded (i.e. displaying the windows player controls like stop and start buttons, auto-playing the multimedia clip).

**!** *Important: Be sure to use the correct object controls to reference and manipulate the WMP object. These object controls have changed significantly since version 6.4. Reference the Microsoft Developer Network for information on the WMP Object Model.*

### 3.3.5 Active Content in Internet Explorer

Based on the patent infringement lawsuit brought against Microsoft by the University of California and Eolas Technologies, several changes in how Internet Explorer interprets Active Content is expected in upcoming versions of the browser. To be prepared for the upcoming changes all USMC courseware shall be developed to accommodate the

changes. Both Microsoft and Macromedia have programming references on how to properly address the issue. Reference the links below for more information.

Changes to the Default Handling of ActiveX Controls (by MSDN)

<http://msdn.microsoft.com/ieupdate/activexchanges.asp>

Macromedia Active Content Developer Center

<http://www.macromedia.com/devnet/activecontent/>

## 4 ACCEPTABLE RESPONSE TIMES

This section outlines the reasonable and acceptable response times for accessing MarineNet content. These are baseline guidelines that must be complied with unless otherwise waived by the Government PM.

### 4.1 Multimedia Response Times

The table below lists the average and maximum response times for multimedia contained within courseware. Average response times are based on existing courseware explicitly developed for MarineNet. Maximum response times represent the upper-limit that a user shall wait to view the media. The maximum response time for static, Flash and Shockwave media is the time it takes from initial activation to the time the media is downloaded and displayed to the user. The maximum response time for streaming media is the time required for the media to start playing. Courseware vendors may not exceed these guidelines without approval from the Government PM.

Type of Media	Average Response Time (seconds)	Maximum Response Time (seconds)
Static Media	~ 2	8
Flash & Shockwave Media	~ 3	8
Streaming Media	~ 5 (Default streaming buffer time)	5 (Default streaming buffer time)

### 4.2 Overall Page Response Times

While multimedia governs the response times for users accessing content, the MarineNet program requires that all content perform within acceptable guidelines. Under normal conditions, any page within the courseware shall display in entirety (multimedia and text) within 10 (ten) seconds. Furthermore, the next button, allowing the user to progress to the next screen, shall be active and effective within 5 (five) seconds without delay for the entire page to load. Note: Courseware navigation menus or frames shall not be considered in the time a page takes to load.

Courseware vendors may not exceed these guidelines without approval from the Government PM.

## 5 CONTENT PROGRAMMING

### 5.1 Static Content

The only acceptable static content format is HTML/JavaScript. Content may include Cascading Style Sheets (CSS), layers, etc., to provide interactivity within the content. XML or XHTML is not permitted at this time.

### 5.2 Dynamic Content

The use of Active Server Pages (ASP) is permitted to provide dynamic interactions between the courseware and the LMS. However, ASP is not permitted to provide general user interfacing, graphic manipulation, or unnecessary session variables. Cold Fusion (CFM), common gateway interface (CGI), and Java Server pages (JSP) are not permitted under any circumstances.

### 5.3 Screen Sizes and Programming

Content developers shall maximize all screen sizes to provide the best overall learning experience for the user. All screens shall be developed for proper display for the standard video screen resolution of 1024x768. Because browser toolbars, scrollbars, and various taskbars consume some of the overall screen size, the content should not exceed 1024 pixels wide or 740 pixels high. More information is provided in section 6.2.

Screen resizing and pop-ups are permitted when documented in the storyboards approved by the Government PM.

### 5.4 Content Packaging

MarineNet content developed to interface with the LMS for tracking purposes shall provide the required AICC course structure files and SCORM meta-data files. Both, the AICC and SCORM meta-data files, are required. The MarineNet LMS currently does not support the SCORM 1.2/1.3 meta-data files therefore all SCORM courses are loaded into the LMS as AICC courses. The SCORM meta-data files are still required to prepare for the future upgrade of the MarineNet LMS as well as to be used with other LMSs.



*Important: Vendors developing toward the SCORM 1.2 specifications shall develop both the SCORM imsmanifest meta-data files as well as the AICC import files identified in this section. Content vendors should contact their Government PM for clarification on what is required for a specific development effort.*

Appendix A provides detailed examples and best practices for the development of the AICC and SCORM meta-data files.

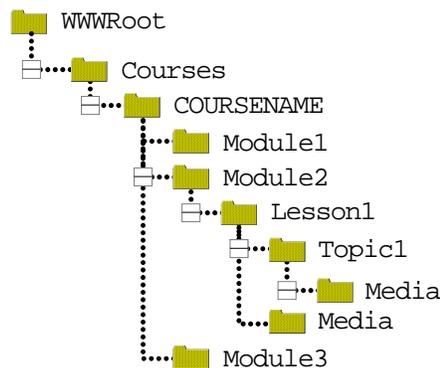
## 5.5 Directory Structure

Courseware must conform to the established directory structure format in order to be distributed to the remote content servers.

Some basic principles dealing with directory structure include:

- The course should be maintained in a single directory.
- All media should be contained within a 'media' subdirectory directly below the directory hosting the web page that references the media. Streaming media and non-streaming media should be collocated in the appropriate media directory.
- Media directories can exist at any branch where content is assigned and media files are required.
- Directory hierarchy should be limited to five (5) levels below the course.

Example



## 5.6 File and Directory Naming Conventions

Courseware file and directory names shall be no longer than twenty (20) characters long; plus the appropriate file type extensions. Courseware file and directory names may not include special characters that can interfere with the distribution of content to the distributed MarineNet Content Servers. These restrictions include:

- No spaces within file or directory names
- No URL-encoded values using the % value. Ex. %20, %09, %0A

## 5.7 Initial Launch File

MarineNet-accessible content shall be launched from a file in the course's home directory. The initial launch file shall be named *index.htm*. The *index.asp* filename may be used when ASP technologies are required.

## 5.8 Acceptable Software Programming Tools and Aids

MarineNet-accessible content has been developed using numerous programming tools and aids. The use of these tools is permitted; however, the Government does not accept

any limitations imposed by these tools, such as lack of user flexibility or requirement for additional plug-ins or players. Such tools include, but are not limited to:

- Macromedia Dreamweaver
- Macromedia HomeSite
- Microsoft FrontPage
- Microsoft Visual Studio
- OutStart Evolution



*Important: Use of software programming tools and aids are permitted; however, the Government does not accept any limitations in instructional design or programming for using these tools. It is recommended that the content vendor inform the Government prior to using such tools to receive more detailed guidance and acceptance criteria.*



*Important: Use of Click2learn Toolbook is not permitted under any circumstances.*

## 5.9 Mobile Code

This section outlines the acceptable and unacceptable use of mobile code as directed by Department of Defense (DoD) Memorandum, Policy Guidance for use of Mobile Code Technologies in DoD Information Systems, dated November 11, 2000.

### 5.9.1 Mobile Code Category Descriptions

Category 1 mobile code technologies exhibit a broad functionality, allowing unmediated access to workstations, host, and remote system services and resources. Category 1 mobile code technologies have known security vulnerabilities with few or no countermeasures once they begin executing.

Category 1 mobile code may be used in DoD information systems only when the mobile code is *signed* with a DoD-approved PKI code signing certificate and the mobile code is obtained from a trusted source.

Category 2 mobile code technologies have full functionality, allowing mediated or controlled access to workstations, host, and remote system services and resources. Category 2 mobile code technologies may have known security vulnerabilities but also have known fine-grained, periodic, or continuous countermeasures or safeguards.

Category 2 mobile code may be used in DoD information systems if the mobile code is obtained from a trusted source over an assured channel. In addition, *unsigned* Category 2 mobile code, whether or not obtained from a trusted source over an assured channel, may be used if it executes in a constrained environment without access to local system and network resources (e.g., file system, Windows registry, network connections other than to its originating host).

Category 3 mobile code technologies support limited functionality, with no capability for unmediated access to workstations host, and remote system services and resources.

Category 3 mobile code technologies may have a history of known vulnerabilities, but also support fine-grained, periodic, or continuous security safeguards. Category 3 mobile code technologies may be used in DoD information systems.

### 5.9.2 Mobile Code Designations

The following technologies are designated Category 1:

- ActiveX Windows Scripting Host, when used to execute mobile code
- Unix Shell Scripts, when used as mobile code
- DOS Batch Scripts, when used as mobile code

The following technologies are designated Category 2:

- Java applets and other Java mobile code
- Visual Basic for Applications (VBA)
- LotusScript
- PerfectScript
- Postscript

The following technologies are designated Category 3:

- JavaScript (include Jscript and ECMAScript variants)
- VBScript
- Portable Document Format (PDF)
- Shockwave/Flash

## 5.10 Code Reuse and Open Source

The reuse of code and use of open source programming is encouraged, where appropriate. No courseware may directly rely, in run-time, on another course or any other external source other than that contained within the course's subdirectory. The use of open source code and reuse of code from another development may not impact the Government ownership of any or all source and compiled code delivered to the Government.



*Important: The use of open source code and reuse of code from another development may not impact the Government ownership of any or all source and compiled code delivered to the Government. Furthermore, the Government does not permit the use or reuse of code that includes restrictions, conditions, or other limitations on ownership, warranty, free distribution or licensing.*

## 6 INTERACTIONS WITH THE MARINET LMS LEARNING MANAGEMENT SYSTEM (LMS)



*Important: This section uses the term 'MarineNet LMS' to refer to the USMC's current LMS. The use of the generic 'LMS' term is used to refer to "any SCORM conformant LMS."*



*Important: This section discusses the implementation of SCO-based end of course (EOC) (ie. competency) exams. The traditional 'EOC tests' term refers to the testing component provided by the LMS. Testing provided within the SCO is referred to as inline or SCO-based EOC tests. The term 'tests' can refer to either type of EOC test.*

### 6.1 Relationship between Courseware and the MarineNet Course Catalog

The MarineNet LMS is responsible for overseeing the enrollment, tracking, and completion reporting of all courses. The MarineNet LMS operates a course catalog that describes each course hosted by MarineNet as well as recording the interaction protocol (i.e. AICC or SCORM) between the LMS and the courseware.

High- and low-bandwidth versions of a course shall operate as a single course; thus having a single course catalog entry. Progress tracking within a course shall be stored in the MarineNet LMS corresponding to the course number, regardless of the high- or low-bandwidth version. All content bookmarks stored in the LMS shall be interoperable between the high- and low-bandwidth versions.

All assignable units (AU) and sharable content objects (SCO) must be identified in the course catalog as a single course. Thus, no multi-AU or multi-SCO courses are permitted in MarineNet.

### 6.2 Courseware Launch Window Sizing

The MarineNet LMS will create a new window that will host the courseware. The courseware window will be automatically repositioned relative to the user's top left corner of their desktop. The size of the courseware window will be 1024 pixels wide by 740 pixels high. The content frame will be the only visible frame within the courseware window and is permitted to use all of the available space within the courseware window. The resizing of the window or the opening of a new primary courseware window is not permitted without approval by the Government PM. Pop-up windows are permitted to provide feedback or for content clarification or remediation.

### 6.3 Interfacing with the SCORM-conformant MarineNet LMS

Interactions between courseware and the MarineNet LMS are based on the SCORM 1.2 specification. The MarineNet LMS provides courseware with a distributed SCORM environment that provides all of the mandatory SCORM functions. MarineNet's distributed SCORM environment incorporates both the AICC HACP and SCORM 1.2 specification. In MarineNet, all courseware is launched as an AICC course using the standard session identification variables located in the URL of the course launch link. If the course is a SCORM-conformant course, then the course launch link that is called from the LMS will also identify a SCORM Relayer.

The SCORM Relayer is part of the LMS and will be responsible for declaring the SCORM API Adapter as well as capturing the AICC variables passed in the URL. The SCORM Relayer is a multi-frame window that is responsible for relaying all SCORM transactions with the centralized LMS as well as launching and interfacing with the SCO. The SCORM Relayer is launched from the distributed courseware servers therefore both the SCORM API Adapter and the SCO are operating within the same Java security sandbox.

The SCORM Relayer and SCORM API Adapter are both launched and configured by the LMS, therefore all SCOs launched by MarineNet will not deviate from the SCORM specification. The SCO will be launched within the content frame of the SCORM Relayer and will be able to consume the entire frame with content as well as interact with the SCORM API Adapter that is located in the second frame of the same window, which is invisible to the user based on a pixel height of zero.

 *Important: Since the courseware window is a frame set that contains a frame that is owned by the LMS (specifically the SCORM Relayer), the SCO must be careful when using `document.top`, `document.all` and other similar relative object addressing since they may reference variables outside the scope of the content frame. Note: The name of the frame that the LMS will launch the content into is "content."*

Because of the implementation of the SCORM Relayer and the limitations of the baseline MarineNet LMS, multi-SCO content is not feasible. Thus, all SCOs must be identified as an individual course in the MarineNet course catalog. Individual courses may be identified as belonging to a single curriculum, however, a student's transcript will record the individual progress of each course once the student is enrolled in the curriculum.

#### 6.3.1 SCORM Limitations in the MarineNet LMS

MarineNet's upgrade to include the ability to host SCORM-conformant courseware does not guarantee that MarineNet will be capable of interfacing with all SCOs. MarineNet only implements the basic SCORM functionality and there are more subtle limitations even within the basic functionality supported by MarineNet.

### 6.3.1.1 Supported SCORM Functions

This section clarifies which SCORM functions are minimally supported by MarineNet and identifies all limitations inherent to the functions supported.

The following functions are supported by MarineNet:

- LMSInitialize
- LMSFinish
- LMSGetValue
- LMSSetValue
- LMSCommit
- LMSGetLastError
- LMSGetErrorString
- LMSGetDiagnostic

All of the functions listed above are fully supported by MarineNet. Refer to the SCORM 1.2 specification for documentation on each of the functions.

### 6.3.1.2 Supported SCORM Data Elements

This section clarifies which SCORM data elements are minimally supported by MarineNet and identifies all limitations inherent to the data elements supported.

The following data elements are supported by MarineNet:

- **cmi.core.student\_id**
  - A value used to uniquely identify the user. The value returned by the LMS is not the user's Social Security Number or the customer id assigned by the LMS.
- **cmi.core.student\_name**
  - The student's name. Note that the format of the name is not standardized and is passed as a single string.
- **cmi.core.lesson\_location\***
  - The bookmark value that is stored/retrieved as a string.
- **cmi.core.lesson\_status**
  - The SCO's overall status (Incomplete, Complete, Passed, Failed)
- **cmi.core.score.raw**
  - The user's score at the SCO level.
- **cmi.core.total\_time**
  - The total time the user has spent in the course throughout all sessions.
- **cmi.core.credit**
  - The Boolean (credit/no-credit) value used to identify if the course has a test assigned, thus determining if the student is required to demonstrate competency of the material.

- **cmi.core.score.\_children**
  - SCORM specific mandatory element. LMS responds with ‘cmi.core.score.raw’.
- **cmi.core.entry\***
  - Identifies how the user last exited the course.
- **cmi.core.exit\***
  - Used by the SCO to tell the LMS how the user is exiting a specific session.
- **cmi.core.session\_time**
  - The time the user has spent in the active session. Used to accumulate total time.
- **cmi.suspend\_data\***
  - General database field stored as a string value and can be used to store any progress type data required by the SCO.
- **cmi.launch\_data\***
  - Read-only data stored as a string value that can be passed to the SCO.
- **cmi.objectives.\_children**
  - SCORM specific mandatory element. LMS responds with ‘cmi.objectives.n.id’, ‘cmi.objectives.n.score’, and ‘cmi.objectives.n.status’
- **cmi.objectives.n.id**
  - Global identifier for the objective. Must match AICC objective identifier in order for the LMS to display the title of the objective in the LMS.
- **cmi.objectives.n.score.raw**
  - The objective’s overall score.
- **cmi.objectives.n.status**
  - The objective’s overall status.

All of the data elements listed above are supported by MarineNet, however the elements identified with asterisks include some limitations beyond those documented in the SCORM 1.2 specification. Any calls to data elements not listed above will return a *not implemented* error message.

### **Limitations of the cmi.core.lesson\_location Data Element**

The SCORM specification identifies that the lesson\_location data element be capable of storing a string of length 255 characters. However, MarineNet limits the lesson\_location data element to only 250 characters. Any characters beyond the first 250 will be discarded.

### **Limitations of the cmi.core.entry Data Element**

The first time that a SCO is launched from the MarineNet LMS the cmi.core.entry data element will be set to “ab-initio”. Every subsequent session will have a cmi.core.entry value of “empty string”. The MarineNet LMS will not ever produce a value of “resume.”

### **Limitations of the *cmi.core.exit* Data Element**

There is no capability to store any data in the *cmi.core.exit* data element. A call to store a value in the *cmi.core.exit* data element will always result in a *success*, however no data will ever be stored in the LMS database.

### **Limitations of the *cmi.suspend\_data* Data Element**

The SCORM specification identifies that the *cmi.suspend\_data* data element be capable of storing a string of length 4096 characters. However, MarineNet limits the *cmi.suspend\_data* data element to only 2000 characters. Any characters beyond the first 2000 will be discarded.

### **Limitations of the *cmi.launch\_data* Data Element**

The SCORM specification identifies that the *cmi.launch\_data* data element be capable of storing a string of length 4096 characters. However, MarineNet limits the *cmi.launch\_data* data element to only 250 characters. Any characters beyond the first 250 will be discarded.

## 6.3.2 Additional SCORM Development Guidelines

This section outlines the required use of SCORM functionality in all USMC-developed SCOs. The primary reason for these guidelines are to provide similar functionality in both operations of the SCOs as well as how they are reported to the LMS and what the user sees within the LMS.

### *6.3.2.1 Initial Launching of the SCO*

Several tasks are required of the SCO when it is launched for the first time by a user. These tasks are required to initialize the SCO so that the MarineNet LMS may properly display user progress. The SCO shall identify an initial launch by querying the *cmi.core.entry* data element. Whenever *cmi.core.entry* is set to “ab-initio” then it can be assumed that this is the first time the user has launched the SCO.

### *6.3.2.2 SCORM Course/SCO Status*

SCORM’s *cmi.core.status* data element identifies the student’s overall progress in the SCO. The possible values are “not attempted”, “incomplete”, “passed”, “failed”, and “browsed.” When the SCO is launched for the first time *cmi.core.status* is set to “not attempted” and should immediately be changed to “incomplete.” A SCO may not set *cmi.core.status* to “not attempted” at any time. When *cmi.core.status* is set to “browsed” the MarineNet LMS will not display any sub-course (e.g. Objective statuses) to the user, however the Marinenet LMS will continue to provide all existing functionality to the SCO.

The use of “complete”, “passed” and “failed” must include error checking so as to not overwrite any previous statuses. The value “complete” should be used to indicate that the user has finished reviewing all of the information within the SCO or objective. The

determination of what “finished reviewing” constitutes will be determined by the Government program manager. Some SCOs or objectives may have the “complete” based on the taking of a test within the SCO or by simply accessing the last page of the SCO or objective.

**!** *Important: If the course does not have a corresponding test, once the SCO stores a `cmi.core.status = "complete"` the MarineNet LMS will automatically delete the user's enrollment. This is because there is no test and the "complete" value indicates that the student has finished reviewing all of the material. If the course does have a corresponding test then the enrollment will persist and will not be deleted until a "pass" or "fail" is stored.*

Since the MarineNet LMS will automatically delete the enrollment once a “pass” or “fail” is stored in `cmi.core.status`, there is little chance that the SCO will have the opportunity of overwriting it with another value. However, there is the possibility that the user may continue to browse the content. Therefore, a SCO shall never overwrite a “pass” or “fail” status. If a course development effort wishes to permit the student multiple attempts to pass an internal test then the SCO should not store a “fail” value until the user has exhausted all attempts.

### 6.3.2.3 Normalization of the Score Value

SCOs storing values into `cmi.core.score.raw` shall only store normalized values between 0 and 100. The score value will be displayed to the student as a percentage, with the total possible score being 100%. The LMS will also determine if the student has passed or failed the SCO by comparing the score to the mastery score stored in the LMS. The LMS will only update the status to passed or failed if the SCO stores a score value, sets the SCO status to ‘completed’ and the LMS has a mastery score for that SCO.

The score values of objectives will also be displayed to the user within the LMS therefore, it is also required that all objective scores be normalized between 0 and 100.

### 6.3.2.4 SCORM Lesson Objectives

SCORM courses using objectives must:

1. Define the objectives in the AICC import files
2. Update the status of all objectives to “not attempted” upon initial launch of the SCO
3. Update the status to “incomplete” once the user opens any part of the objective
4. Update the status to “completed” whenever the user finishes an objective, and
5. (OPTIONAL) Update the status to “passed” or “failed” whenever the user completes an in-line quiz/test for that objective.

The initialization of the objectives upon initial launch is required to allow the LMS to display all of the objectives associated with the course regardless of whether the user has accessed the objective. Once the course has identified that the user has launched the course for the first time, then the course shall set the objective ID and statuses for each of the objectives. The objective ID set must correspond to the IDs documented in the AICC import files otherwise, the objective titles will not be displayed in the user's module progress view in the LMS and they will be replaced with the objective ID.



*Important: When initializing objectives be sure to create them in order so that they appear in the correct order in the MarineNet LMS.*

#### 6.3.2.5 Objective Identifiers

The use of both SCORM's `cmi.objectives.n.id` and AICC's `objective_id` may be confusing since the data types are significantly different. The MarineNet LMS's data type of the objective identifiers is based on the SCORM data type of 255 characters without any white spaces or special characters. There is no requirement to use the AICC mechanism for naming objectives, however the SCO must use some logical naming mechanism such as "obj\_1, obj\_2" or "j\_01, j\_02". The actual title shall be the only descriptor that is shown to the user so the IDs used in the LMS's AICC tables must match those being referenced by the SCO. It will not be possible for the SCO to query the objective title from the LMS since it is not supported by SCORM.

#### 6.3.2.6 SCO Sessions and Attempts

While it is not clearly documented in the SCORM 1.2 specification there is a substantial difference between a SCO session and a SCO attempt. The SCO attempt is the student's "registration" or "enrollment" in the course/SCO and may consist of multiple visits to the SCO. Each visit is considered a session and may retrieve and store information about the student's process within that session or within other sessions during that *attempt*.

When the user leaves the SCO, it is the SCO's responsibility to report to the LMS whether the student is simply terminating that *session* or terminating the entire *attempt*. The distinction is important since SCOs are not able to recall information about previous attempts therefore a termination of the student's attempt may be characterized as the student's dropping of the course. Once the attempt is terminated, the information about the attempt is frozen and cannot be changed.

The `cmi.core.exit` data element is the SCORM mechanism to determine if the student is terminating the attempt or simply terminating the session. Storing a value of "suspend" will inform the LMS that the student is simply terminating that session and expects to return to the SCO at a later time. Storing values of "logout", "empty string", or "time-

out” implies that the student will not be returning and therefore the student’s attempt should be terminated.



*Important: All USMC-developed SCOs shall always record a `cmi.core.exit` value of "suspend" prior to the calling of `LMSFinish()`.*

The MarineNet LMS does not implement the implied intent of the SCORM 1.2 specification with respect to use of `cmi.core.exit` to manage SCO attempts. The SCORM 1.2 specification is expected to be updated to clarify this intent. It is expected that all future SCORM-conformant LMSs will comply. Thus, to reduce future maintenance concerns, all USMC-developed SCOs shall always store a value of “suspend” in the `cmi.core.exit` data element during every visit to the SCO.

## 7 TESTS AND SURVEYS

MarineNet supports the following types of tests and surveys:

- Pre-assessments
- In-line Quizzes
- Practice Tests
- End-of-Course (EOC) tests
- Surveys

### 7.1 Pre-assessments, In-line Quizzes, Practice Tests, and Surveys

Pre-assessments, in-line quizzes, practice tests, and surveys must be integrated within the courseware. Results of these components may be reported to the MarineNet LMS for tracking and reporting.

The `lesson_status` variable should be updated to record passing and failing statuses as well as course completions whether through pre-tests or internal EOC tests. All other tracking information must be passed in the `lesson_location` variable or through the generic LMS/AICC Communication Capability. Refer to Section 6 for more information on MarineNet LMS interfaces.

### 7.2 MarineNet LMS End-of-Course (EOC) Tests

EOC tests can be located within a course with the results passed to the MarineNet LMS, or EOC tests can be created within the MarineNet LMS. This section documents the requirements for importing an EOC test into the MarineNet LMS.

Only one EOC test can be associated with each course in the MarineNet LMS. EOC tests may either be manually entered into the MarineNet LMS via the graphical-user-interface (GUI) or automatically imported through the database import tool. Course vendors shall format tests to allow for the automatic importing of tests into the MarineNet LMS. Courseware vendors shall submit EOC tests for insertion into the LMS in a single Microsoft Excel workbook with four worksheets. The first worksheet will contain the learning objectives, the second worksheet will contain the questions, the third worksheet will contain the responses associated with the questions, and the fourth worksheet will contain administrative information and special instructions. To ensure the EOC tests can be imported properly, all of the EOC test data must be formatted as described in Sections 7.2.1 – 7.2.3.

Vendors shall also provide EOC tests in Microsoft Word format with questions grouped by Learning Objectives.

#### 7.2.1 Learning Objectives Worksheet

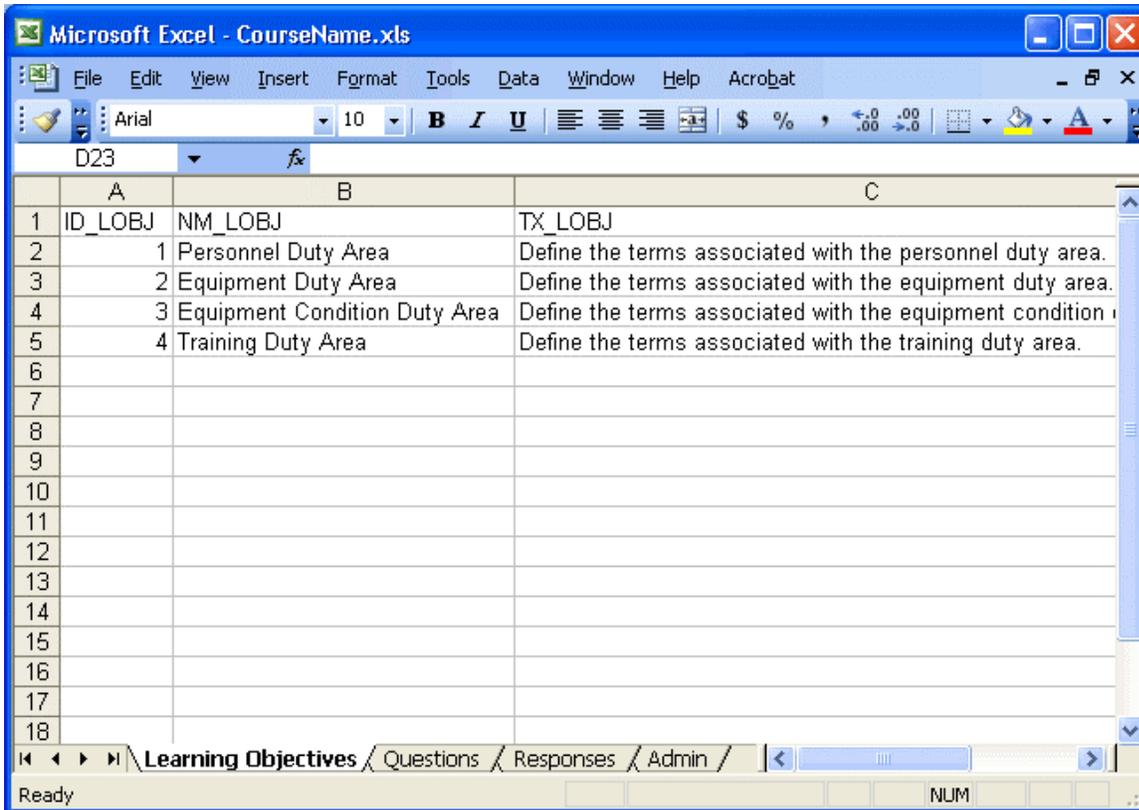
The first worksheet of the EOC Excel workbook shall be named “Learning Objectives” It will list the learning objectives for the test being created. This Excel worksheet contains

the learning objectives associated with the test. The required columns are described in the table below.

Field Name	Datatype	Field Size	LMS Database	Explanation
ID_LOBJ	Integer	N/A	Integer	Learning Objective ID: Unique ID of the Learning Objective.
NM_LOBJ	String	254 chars	254 chars	Learning Objective Name: Title of the Learning Objective.
TX_LOBJ	String	2000 chars	2000 chars	Learning Objective Text: Description of the Learning Objective

Below is an example of a learning objectives worksheet with four learning objectives. Note that the first line lists the header.

*Example*



### 7.2.2 Test Questions Worksheet

The second worksheet of the EOC Excel workbook shall be named “Questions”. It will list the questions for the test being created. This Excel worksheet contains the questions associated with the test. The required columns are described in the table below.

Field Name	Datatype	Field Size	LMS Database	Explanation
ID_LOBJ	Integer	N/A	Integer	Learning Objective ID: Unique ID of the Learning Objective that this question is associated with. Ties to ID_LOBJ from Learning Objectives file.
ID_QUESTION	Integer	N/A	Integer	Question ID: Unique ID of the question.
TX_QUESTION	String	2000 chars	2000 chars	Question Text: The actual question text.
NO_QUESTION_GRADE	Integer	N/A	Integer	Question Grade: The number of points the question is worth. This value should be 100
CD_RESPONSE_TYPE	String	10 chars	10 chars	Response Type: The type of question. Possible values are: <ul style="list-style-type: none"> <li>• <b>MC</b> Multiple Choice (Used for questions with more than one correct answer. Different weight values can be assigned to the answers to enable the user to receive partial credit)</li> <li>• <b>SC</b> Single Choice (Used for questions with several responses available, but only one correct answer)</li> </ul>
TX_GRAPHIC_FILE	String	254 chars	254 chars	Graphic File: Relative pathname that leads to the location of a graphic file associated with the question. If there is no graphic file, leave this field blank.
TX_RESPONSE_DESC	String	2000 chars	2000 chars	Response Description: The correct answer for the question

Below is an example of a test questions worksheet with ten questions associated with it. Note that the first line lists the header. All of the questions are since choice (SC) questions.

**Example**

	A	B	C	D	E	F	G
1	ID_LOBJ	ID_QUEST	TX_QUESTION	NO_QUEST	CD_RESP	TX_GRAP	TX_RESPONSE_DESC
2	1	1	The two types of clothing	100	SC		basic and standard.
3	1	2	A Marine enlisted in the	100	SC		13-Nov-94
4	1	3	To receive demolition pa	100	SC		demolish underwater ob
5	2	4	To determine which pha	100	SC		aviation service entry da
6	2	5	Establishment of legal r	100	SC		enters into the service.
7	2	6	Taxable wages for Fede	100	SC		basic pay.
8	3	7	Use the chart for the fol	100	SC	/MARTest	\$18.00
9	3	8	Use the chart for the fol	100	SC	/MARTest	\$62.50
10	4	9	Enlisted members may	100	SC		permission to mess sep
11	4	10	A MGySgt with 21 yrs c	100	SC	/MARTest	\$2,924.10
12							
13							
14							
15							
16							
17							
18							

**7.2.3 Valid Responses Worksheet**

The third worksheet of the EOC Excel workbook shall be named “Responses”. It will list the responses for the test being created. This Excel worksheet contains the responses associated with each question. The required columns are described in the table below.

Field Name	Datatype	Field Size	LMS Database	Explanation/Example
ID_QUESTION	Integer	N/A	Integer	Question ID: Unique numeric ID associated with the question that ties back to the ID_QUESTION field from the Questions file.
TX_RESPONSE	String	2000 chars	2000 Chars	Response Text: Each answer associated with the question will appear in a separate row in the Valid Responses file. <b>Note: One of these must exactly match the TX_RESPONSE_DESC field in the Test Questions file.</b>
NO_SEQ	Integer	N/A	Integer	Sequence Number: Integer that denotes the order in which the answer should appear during the test. <b>Note: The LMS will randomize this order.</b>
RT_WEIGHT	Real	N/A	Real	Response Weight: The amount of points the response is worth. If the response is incorrect, it should be worth 0 points. If there is only one correct response, it should be worth 100 points. If multiple responses are correct, they should each have point values associated with them, and the total value of the responses should not exceed 100 points.

Below is an example of a valid responses worksheet with ten sets of valid responses associated with it. Note again that the first line lists the header.

**Example**

	A	B	C	D	E	F	G	H	I
1	ID	QUESTION	NO_SEQ	RT_WEIGHT					
2	1	replacement and s	1	0					
3	1	basic and supplier	2	0					
4	1	basic and standar	3	100					
5	1	beginners and sta	4	0					
6	2	8-May-94	1	0					
7	2	13-Nov-94	2	100					
8	2	9-May-94	3	0					
9	2	14-Nov-94	4	0					
10	3	participate as stud	1	0					
11	3	demolish underwa	2	100					
12	3	experiment with m	3	0					
13	3	demolish underwa	4	0					
14	4	HALO service enti	1	0					
15	4	flying service entr	2	0					
16	4	aviation service er	3	100					
17	4	officer date of rank	4	0					
18	5	enters into the ser	1	100					

Importing these three worksheets into the LMS will create a test with the ten questions, the appropriate Learning Objectives of the test, the appropriate responses referenced in the Valid Responses worksheet, and the relationships between the Test and Learning Objectives and the Questions per Learning Objective.

**7.2.4 Administrative Worksheet**

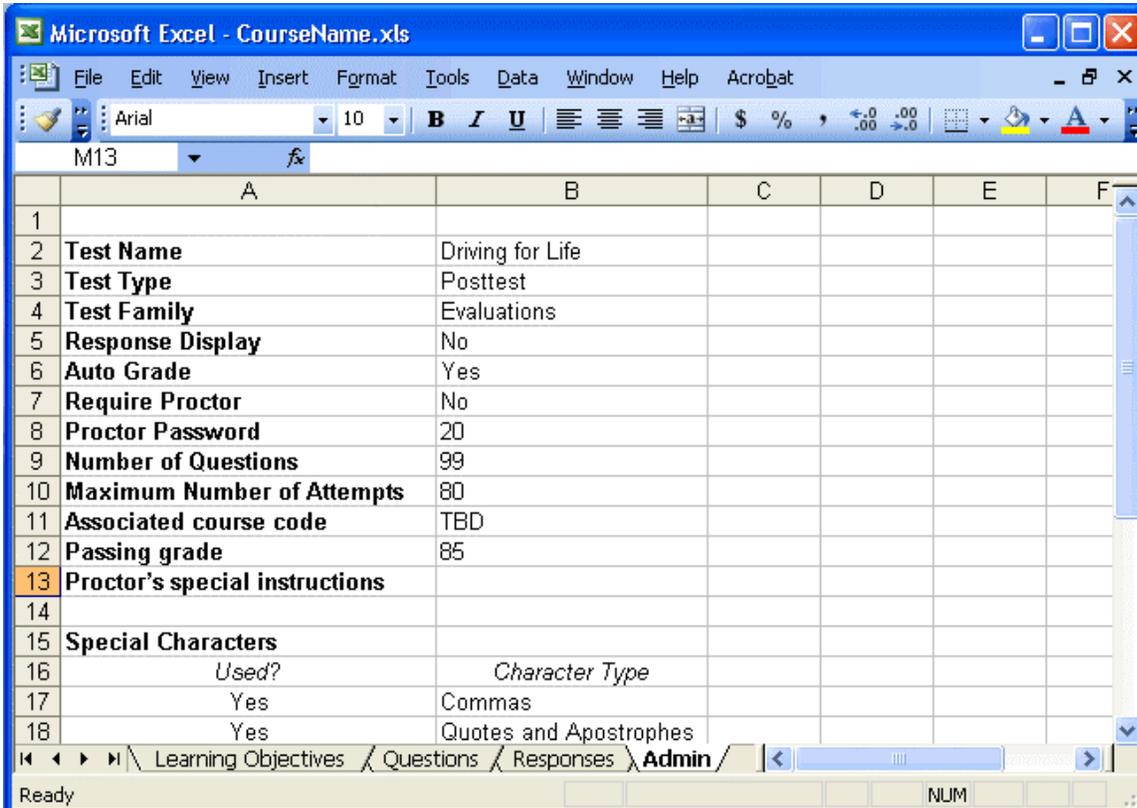
The last worksheet of the EOC Excel workbook shall be called “Admin”. This worksheet will be used to convey administrative information to the LMS administrators. The following information, which can be obtained from the Government PM, shall be included:

- Test Name
- Test Type
- Test Family
- Response Display
- Auto Grade
- Require Proctor
- Proctor Password
- Number of Questions
- Maximum Number of Attempts
- Associated course code
- Passing grade

- Proctor’s special instructions

This worksheet also shall list any special instructions for importing the test data. The main purpose is to prevent delimiter problems during the database import process. At a minimum, any special characters used within the worksheets shall be identified as well as any special instructions for the images.

**Example**



## **8 FUTURE TRENDS**

The MarineNet system has undergone numerous improvements since it first went online in 1998, and over the next few years more changes are expected. This section outlines some of the possible future changes to the system that may affect courseware. Content vendors are encouraged to review this section to help guide development and maximize the lifetime of the content.

### **8.1 Client Software Baselines: Browsers and Plug-ins**

MarineNet users aboard Marine Corps installations are required to comply with NMCI and/or local Marine Corps standards regarding desktop software. These standards are incorporated into the MarineNet Software Baseline. Marine Corps users are prevented from installing or reconfiguring software applications (including web browsers, plug-ins, and operating system components). With new versions of software constantly emerging, the software baselines will continue to be updated. CCE maintains all active courseware to ensure that all are compatible with updated software baselines.

### **8.2 Internet-based Content: LMS Tracking and Performance**

MarineNet began the process of providing Internet-accessible services in 2002. The offering of services via the Internet imposes some potential problems for content vendors.

First, content must be capable of being delivered to the user in an efficient manner. Content must be able to reach the user by traversing several firewall boundaries without excess delay. This means that content must consider external firewall policies and be optimized to support lower bandwidth conditions.

Secondly, content developers must consider that the Internet user's workstations will be configured in various ways. While all Marine Corps computers can be expected to operate standardized software loads, Internet users may not have the latest plug-ins or applications. Content vendors developing content to target Internet users must work with the Government PM to define the assumptions of what software will be loaded on Internet user's workstations and what can be expected to be loaded by the user if the required software is not installed or up-to-date.

### **8.3 AICC, SCORM, and other LMS Interface Standards**

While MarineNet is currently based on the AICC HACP standard, the program is transitioning to the Sharable Content Object Reference Model (SCORM) specification. The complexity of this transition is based largely on the restrictions of operating in a distributed environment. The Marine Corps is actively working with ADL to investigate strategies for altering the SCORM specification to include the capability for commercial LMSs to inherently support a distributed content hosting architecture. As of yet, only a few LMS vendors have recognized the issue and have begun providing support for enterprise distributed learning architectures. The current MarineNet LMS is built on a legacy commercial LMS that does not inherently support the SCORM interactions with distributed content. The MarineNet system provides an intermediary solution by

providing a SCORM Relayer that translates SCORM transactions into AICC HACP. This is not a complete solution, however it will allow for minimal conformance until MarineNet is able to transition to a new LMS.

#### **8.4 Legacy MarineNet Content**

MarineNet will continue to host content that does not conform to the specific guidelines contained in this document as standards and technology implementations change over the life of a course. Legacy content shall be updated on a periodic basis through the course lifecycle, based on the priorities assigned by the CCE.

## APPENDIX A. AICC AND SCORM META-DATA FILES

This appendix outlines the minimum requirement for the AICC and SCORM meta-data files. For all SCORM courses, both the AICC and SCORM meta-data files are required.

### AICC META-DATA FILES

AICC meta-data files are used by the MarineNet LMS to import AICC and SCORM course information. The AICC meta-data files do not offer the same amount of information about the content as the SCORM meta-data files; however the current MarineNet LMS does not support the importing of SCORM meta-data files.

Information on these files and more details about what they should contain may be found in Section 6 of the AICC document, CMI001 (<http://www.aicc.org/pages/down-docs-index.htm>).

The four required AICC meta-data files are:

- file.CRS
- file.DES
- file.AU
- file.CST

The following sections deal with each of these files in some detail.



*Important: All meta-data files shall be provided with the courseware for testing and acceptance. All course meta-data files shall be located in the root directory of the course.*

### ***AICC meta-data file -- .CRS***

This file contains information about the course as a whole. It offers information that relates to more than just a single element of the course.

This file should be constructed in the Group/Keyword style, like a Microsoft Windows INI file.

**Example**

```
[Course]
Course_Creator = Vendor Name
Course_ID = 0215
Course_System = USMC Style Guide Version #
Course_Title = Terrorism Awareness (Web)
Level = 1
Max_Fields_CST = 2
Total_AUs = 1
Total_Blocks = 1
Total Objectives = 4
Version = 3.0
[Course_Behavior]
Max_Normal = 1
[Course_Description]
This course covers the following topics: terrorism basics,
terrorist groups by region, terrorism and you, and U.S. policy.
Designed to increase the awareness of Marines and their families
to reduce their vulnerability to terrorism when deployed to a
high terrorism threat region.
```

For more information on this file, see Section 6.1 of the AICC document, CMI001.

***AICC meta-data file -- file.AU***

This file contains information about each assignable unit (AU) in the course, including the data needed to launch the unit. All MarineNet content shall be configured as one AU.

This file should be constructed as a table, in comma delimited ASCII.

**Example**

```
"system_id", "type", "command_line", "Max_Time_Allowed", "time_limit
_action", "file_name", "max_score", "mastery_score", "system_vendor"
, "core_vendor"
"A001", "SCO", , , , "http://www.marinenet.usmc.mil/courses/sconame/i
ndex.htm", , , " USMC Style Guide Version #", ""
```

For more information on this file, see section 6.2 of the AICC document, CMI001.

***AICC meta-data file -- file.DES***

This file contains a complete list of every content element defined in the course. It is used as the basic cross-reference file showing the correspondence of system-generated IDs with user-defined IDs for every element. This file also contains text descriptions for the content elements. Content elements include Assignable Units, Blocks, Objectives, and Complex Objectives.

This file should be constructed as a table, in comma delimited ASCII.

**Example**

```
"System_ID", "Developer_ID", "Title", "Description"  
"A001", "USMC_0215AO_A001", "Terrorism Awareness", "Terrorism  
Awareness Course"  
"J0100", "USMC_0215AO_J0100", "Security and the Internet", "To  
introduce the concept of Internet security"  
"J0200", "USMC_0215AO_J0200", "Security threats", "To introduce the  
individual categories of Internet security threat to an  
organization and explain the inevitable consequences"  
"J0300", "USMC_0215AO_J0300", "Secure computing", "To explain how  
to secure clients and applications"  
"J0400", "USMC_0215AO_J0400", "Secure networking", "To explain how  
to secure your network, web platform, and communications against  
hostile attacks"
```

For more information on this file, see Section 6.3 of the AICC document, CMI001.



*Important: All objectives must be identified in the .DES file. Courses are not permitted to dynamically create objectives that were not identified in the .DES file.*

**AICC meta-data file -- file.CST**

This file contains data about the structure of the course. It includes all of the assignable units and blocks in the course. The order in which these blocks appear in the file implies (but does not force) an order for presentation to the student.

This file should be constructed as a table, in comma delimited ASCII.

**Example**

```
"Block", "Member"  
"Root", "A001"
```

For more information on this file, see Section 6.4 of the AICC document, CMI001.

## SCORM META-DATA FILES

The SCORM meta-data files provide technical and educational information about the learning objects with particular focus on identifying the educational/learning constructs that the object/course is based on, the technical requirements for interacting with the course, and associated lifecycle information to aid in configuration management.

SCORM meta-data files are used by the MarineNet Meta-data Repository System and by SCORM-conformant LMSs. The current MarineNet LMS does not support the importing of SCORM meta-data files, therefore only the AICC meta-data files are used for importing course/learning object information into the MarineNet LMS.

The SCORM meta-data requirement is based on the Instructional Management System Global Learning Consortium, Inc. (IMS) and follow-on Advanced Distributed Learning (ADL) specifications. Refer to the IMS Meta-data and Content Packaging specifications as well as the ADL SCORM 1.2 Content Aggregation Specifications for more information about the data elements outlined in this section.

### *SCORM Meta-data file – imsmansifest.xml*

Below is an example of an imsmansifest.xml file required for all MarineNet-developed SCORM courses. The imsmansifest.xml file is used to describe the aggregation of SCOs and assets to create an assignable learning object. The MarineNet LMS only supports single-SCO courses; therefore, only one resource should be listed, the SCO.

```
<?xml version="1.0"?>
<manifest identifier="SingleCourseManifest" version="1.1"
xmlns="http://www.imsproject.org/xsd/imscp_rootv1plp2"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1plp2
http://www.imsproject.org/xsd/imscp_rootv1plp2.xsd
http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
http://www.imsproject.org/xsd/imsmd_rootv1p2p1.xsd
http://www.adlnet.org/xsd/adlcp_rootv1p2
http://www.adlnet.org/xsd/adlcp_rootv1p2.xsd">
  <metadata/>
  <organizations default="B0">
    <organization identifier="B0">
      <title>MarineNet Test SCO</title>
      <item identifier="B100" isvisible="true">
        <title>MarineNet Test SCO</title>
        <item identifier="S000001" identifierref="R_S000001"
isvisible="true">
          <title>MarineNet Test SCO</title>
        </item>
      </item>
    </organization>
  </organizations>
  <adlcp:location>sco01.xml</adlcp:location>
  <adlcp:masteryscore>85</adlcp:masteryscore>
</manifest>
```

```

    <resources>
      <resource identifier="R_S000001" type="webcontent" adlcp:scormtype="sco"
href="index.htm">
        <metadata>
          <schema>ADL SCORM</schema>
          <schemaversion>1.2</schemaversion>
          <adlcp:location>sco01.xml</adlcp:location>
        </metadata>
        <file href="index.htm"/>
      </resource>
    </resources>
  </manifest>

```

### ***SCORM Meta-data file – sco01.xml***

Below is an example of an SCO/Asset Meta-data file required for all MarineNet-developed SCORM courses. Every SCO and/or reusable Asset shall have a corresponding meta-data file that describes the object.

```

<?xml version="1.0"?>
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
imsmd_rootv1p2p1.xsd">

  <general>
    <title>
      <langstring xml:lang="en">MarineNet Test SCO</langstring>
    </title>
    <catalogentry>
      <catalog>MarineNet</catalog>
      <entry>
        <langstring xml:lang="en">TBD</langstring>
      </entry>
    </catalogentry>
    <language>en</language>
    <description>
      <langstring xml:lang="en">This course is about demonstrating the use
of metadata for courses developed to be hosted on MarineNet</langstring>
    </description>
    <keyword>
      <langstring xml:lang="en">MarineNet Test SCO</langstring>
    </keyword>
    <keyword>
      <langstring xml:lang="en">MarineNet</langstring>
    </keyword>
    <keyword>
      <langstring xml:lang="en">Metadata</langstring>
    </keyword>
    <keyword>
      <langstring xml:lang="en">example</langstring>
    </keyword>
    <aggregationlevel>
      <source>
        <langstring xml:lang="x-none">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">3</langstring>
      </value>
    </aggregationlevel>
  </general>
  <lifecycle>
    <version>

```

```

    <langstring xml:lang="en">1.0</langstring>
</version>
<status>
  <source>
    <langstring xml:lang="x-none">LOMv1.0</langstring>
  </source>
  <value>
    <langstring xml:lang="x-none">Final</langstring>
  </value>
</status>
<contribute>
  <role>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Publisher</langstring>
    </value>
  </role>
  <centity>
    <vcard>BEGIN:VCARD VERSION:2.1
    N:Marine Corps College of Continuing Education
    ORG:USMC;TECOM1;MCB Quantico
    ADR;DOM;WORK:;3300 Russel Rd;Quantico;VA; 22134-5135;U.S.
    URL:http://www.tecom.usmc.mil/dlc
    END:VCARD</vcard>
  </centity>
  <date>
    <datetime>2003-12-25</datetime>
  </date>
</contribute>
</lifecycle>
<metametadata>
  <metadatascheme>ADL SCORM 1.2</metadatascheme>
</metametadata>
<technical>
  <format>text/html</format>
  <format>text/plain</format>
  <format>image/gif</format>
  <format>image/jpeg</format>
  <format>application/x-shockwave-flash</format>
  <location type="URI">index.htm</location>
  <requirement>
    <type>
      <source>
        <langstring xml:lang="x-none">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Browser</langstring>
      </value>
    </type>
    <name>
      <source>
        <langstring xml:lang="x-none">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Microsoft Internet
Explorer</langstring>
      </value>
    </name>
    <minimumversion>6.0</minimumversion>
  </requirement>
</requirement>

```

```

    <type>
      <source>
        <langstring xml:lang="x-none">ADL</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Browser Plug-in</langstring>
      </value>
    </type>
  </name>
  <source>
    <langstring xml:lang="x-none">LOMv1.0</langstring>
  </source>
  <value>
    <langstring xml:lang="x-none">Macromedia Flash and
Shockwave</langstring>
  </value>
</name>
<minimumversion>5</minimumversion>
</requirement>
<requirement>
  <type>
    <source>
      <langstring xml:lang="x-none">ADL</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Browser Plug-in</langstring>
    </value>
  </type>
  <name>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Windows Media
Player</langstring>
    </value>
  </name>
  <minimumversion>6.4</minimumversion>
</requirement>
</technical>
<educational>
  <interactivitylevel>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">medium</langstring>
    </value>
  </interactivitylevel>
  <typicallearningtime>
    <datetime>0000-00-00T00:08:00</datetime>
  </typicallearningtime>
</educational>
<rights>
  <cost>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">no</langstring>
    </value>
  </cost>
  <copyrightandotherrestrictions>

```

```

    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">no</langstring>
    </value>
  </copyrightandotherrestrictions>
</rights>
<classification>
  <purpose>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Discipline</langstring>
    </value>
  </purpose>
  <description>
    <langstring xml:lang="en">Courseware Developer</langstring>
  </description>
  <keyword>
    <langstring xml:lang="en">Courseware Developer</langstring>
  </keyword>
</classification>
<classification>
  <purpose>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Educational Objective</langstring>
    </value>
  </purpose>
  <description>
    <langstring xml:lang="en">Create metadata files that conform to
MarineNet requirements (Module 1)</langstring>
  </description>
  <keyword>
    <langstring xml:lang="en">Metadata</langstring>
  </keyword>
</classification>
<classification>
  <purpose>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
    <value>
      <langstring xml:lang="x-none">Accessibility
Restrictions</langstring>
    </value>
  </purpose>
  <description>
    <langstring xml:lang="en">508 Compliant</langstring>
  </description>
  <keyword>
    <langstring xml:lang="en">508</langstring>
  </keyword>
</classification>
<classification>
  <purpose>
    <source>
      <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>

```

```
    <value>
      <langstring xml:lang="x-none">Security Level</langstring>
    </value>
  </purpose>
  <description>
    <langstring xml:lang="en">Access Level 2</langstring>
  </description>
  <keyword>
    <langstring xml:lang="en">Access Level 2</langstring>
  </keyword>
</classification>
</lom>
```