S1000D Transformation Toolkit

Version 3.0

9/30/2011

Contents

1. Introduction

1.1.About this document

This documentation covers conceptual, usage, integration and troubleshooting information needed to operate the S1000D Transformation Toolkit, an open source tool that converts S1000D 4.0 and 4.1 data into a SCORM 2004 3rd Edition content package, a mobile web application or Portable Document Format (PDF). The Toolkit is one deliverable from "The Bridge Project". The Bridge Project's objective is to reduce data ownership costs and increase production efficiencies by integrating technical data and training content production in a common source database (CSDB). The Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L)) funded the project from its Reduction in Total Ownership Cost (RTOC) program. Advanced Distributed Learning (ADL) Initiative, Office of the Secretary of Defense for Personnel and Readiness (OSD(P&R)) managed the Bridge Project.

NOTE: Subsequent references to the Toolkit in this document mean the S1000D Transformation Toolkit. Subsequent references to SCORM mean SCORM 2004 3rd Edition. Subsequent references to S1000D include both 4.0 and 4.1 releases.

1.2. The Bridge Project Motivation

The goal to educate and train in rapidly changing technical environments requires information products of many types to be efficiently updated by and distributed to the right people at the right time and in the right format. To reach this goal, organizational practices that have historically delayed training content production until after products and systems are deployed must share common business processes and infrastructure, such as data acquisition, production, management and delivery. Sharing business practices and infrastructure is now possible with the release of the S1000D Technical Data Specification, Issue 4.0. The release provides a data strategy that allows learning data to be expressed in a neutral way that directly ties content to products, components, and doctrines. S1000D is an ideal part of a data readiness solution because:

- S1000D has been widely adopted across programs and countries.
- S1000D uses XML, the very nature of which is neutral, nonproprietary and portable.
- S1000D supports data interchange between programs and vendors.
- S1000D ties each item of data to a system component via a Standard Numbering System (SNS), a set of unique configuration codes.

Currently developers create and maintain technical and training content in applications and databases that are not integrated. This lack of integration negatively affects life cycle costs and readiness in these ways:

- Training development timelines can be long due to developmental and operational system testing schedules. Technical data and training content are both dependent on system testing finalization. Direct and timely data reuse and repurposing before, during and after testing is prevented by unintegrated authoring environments.

- Training content is not traceable to authoritative technical content. Thus a lag time exists between the time engineering change specifications are received and training content is identified and updated.
- Training content is often needlessly duplicated due to a lack of authoritative source and system traceability.
- The costs of development and update of training are not factored into life cycle costs due to lack of standard processes tied to authoritative technical content.

The Bridge Project will provide a collection of tools that will "bridge" the technical data and learning content domains using S1000D as shared data specification. The Bridge tools will:

- Provide an ability to integrate any learning content authoring tool with any authoritative source database.
- Provide an ability to rapidly adjust learning content in response to changing system requirements.
- Optimize the development process for content reuse.
- Facilitate the integration of technical learning content into life cycle support.
- Transform S1000D learning and technical data modules into a SCORM content package, mobile web application and PDF formats.

1.3. S1000D Transformation Toolkit

The Toolkit is an open source framework and a reference implementation supporting the transformation, packaging and viewing of S1000D data into SCORM content package, mobile web application and PDF formats. The functionality offers a consistent method for systems-based training developers to create technical learning content using traceable S1000D asset formats grouped into learning modules and transform the assets and grouped learning modules into one of these formats.

1.4. S1000D Transformation Toolkit Scope

1.4.1. S1000D Support

The Toolkit supports content authored, published and maintained according to S1000D Releases 4.0 and 4.1. Current capability has focused on providing support for learning data modules and the SCORM Content Package Module (SCPM). The viewing stylesheets provided in the Toolkit were designed specifically to transform the Descriptive and Procedural DMs provided in the Bike Sample data. Implementers can extend the framework to allow full support of any data module type by providing their own stylesheets. Additionally, the Toolkit support only covers the transformation of the organizing schemas. New features of the 4.1 Learning Data Modules are not yet supported.

Schemas that are out of scope of this Toolkit release include:

- Crew/Operator
- Fault

- Parts Data
- Maintenance Schedule
- Process Module
- Container
- Wiring Data
- BREX
- Technical Information Repository
- Checklists

NOTE: For the SCORM output, the Toolkit does not provide a comprehensive S1000D content viewer. The viewer application provided with the Toolkit is designed to specifically to support the S1000D markup found in the S1000D Bike Learning Sample. If output content processed by the Toolkit does not render as expected, it may be necessary to customize the viewer rendering transformation files to achieve the desired rendering results.

1.4.2. SCORM Support and Limitations

Only support for SCORM 2004 3rd Edition content package conformant output is provided. The Toolkit provides the ability to produce two different ways to render the assessments in the SCORM content package. One option handles the assessments with Flash templates and the other option transforms the assessments to HTML/JavaScript formats.

The current Toolkit release does not transform S1000D sequencing and navigation features into the SCORM content package. If sequencing and navigation behaviors are applied to the SCORM output, they will not be present in subsequent transformed versions of the same content. These open source features may be added as needed by toolkit users and supported in future releases of the toolkit.

1.4.3. Mobile Web Application Support and Limitations

The mobile web application output is based on the JQuery Mobile Framework 1.0 Beta 3. Currently the mobile web application output is available in two different versions, performance support output and course output. The performance support output is intended to be used in the field to get information while executing a task, so no assessments are included. Data modules with a learnEventCode of "E" will not be included this output. The mobile course output includes all of the learning data modules and technical data modules referred to by the SCPM.

References to ICN files in the data modules that are Flash files (.swf) are included in the mobile web application output. However, not all mobile devices (e.g., Apple products) support Flash so these files will not render on all devices.

See also:

- http://jquerymobile.com/
- http://jquerymobile.com/gbs/ (contains list of devices supported by JQuery Mobile)

1.4.4. PDF Support and Limitations

The PDF outputs are produced using the Flying Saucer open source project. Two output formats for PDFs are supported. The first output format is a student format that contains the text and graphic content. The other output format for the PDF is an instructor format. The only difference between the instructor version and the student version is that the instructor version contains the answers for the assessments. Flash files in the data modules are not supported in the PDF file. Also active user interfaces such as drop-downs are not supported.

See also:

- http://code.google.com/p/flying-saucer/

1.5.About S1000D

S1000D is an international specification for the procurement and production of technical publications. It is an SGML/XML standard for preparing, managing, and exchanging equipment maintenance and operations information. It was initially developed by the AeroSpace and Defence Industries Association of Europe (ASD) for use with military aircraft. The standard has since been modified for use with land, sea, and commercial equipment. S1000D is copyrighted by the S1000D Council, which includes board members from ASD, the United States' Aerospace Industries Association (AIA), and the Air Transport Association (ATA). It is maintained by the S1000D steering committee, which includes national industry and defense representatives from most of the countries currently using the standard.

See also:

- http://www.S1000D.org

1.6.About SCORM

The Sharable Content Object Reference Model (SCORM®) integrates a set of related technical standards, specifications, and guidelines designed to meet SCORM's high-level requirements—accessible, interoperable, durable, and reusable content and systems. SCORM content can be delivered to your learners via any SCORM-compliant Learning Management System (LMS) using the same version of SCORM.

See also:

- http://www.adlnet.gov/Technologies/scorm
- http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/Previous%20Versions/index.aspx

1.7.About Mobile Web Applications

A mobile web application can be hosted on any web server and accessed through a device's web browser. This Toolkit produces the mobile web application to run across platforms and devices using open web technologies like HTML, CSS and JavaScript. It uses the JQuery Mobile

framework to provide a UI that is similar to native application UIs and provides touch interactions such as swiping, tapping, pinching and rotating.

See also:

- http://www.w3.org/TR/mwabp/
- http://www.w3.org/2011/02/mobile-web-app-state.html

1.8.About PDF

Portable Document Format (PDF) is an open standard for document exchange.

See also:

- http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/PDF32000_2008.pdf

1.9.Toolkit Users

The primary Toolkit implementer and user communities will be CSDB vendors, LMS vendors, content development tool vendors and application developers. The CSDB vendors can integrate the Toolkit into their CSDB application to produce the SCORM content package, mobile web application and PDF formats. LMS vendors can integrate the Toolkit into their applications to fetch and import S1000D training content from a CSDB. Content development tool vendors can incorporate the Toolkit for developmental and testing purposes. Independent vendors will be able to create applications that fetch content from any CSDB and create the SCORM content package, mobile web application and PDF formats.

1.10. Licensing

The Toolkit is licensed for use, at the user's election, under the GNU Lesser General Public License, under the Eclipse Public License v 1.0 or the Apache Software Foundation License v2.0.

If, at the time of use, the Project Management Committee, consisting of the Source Forge project team members, has designated another version of these licenses or another license as being applicable to the Toolkit, user may select to have its subsequent use of the Toolkit governed by such other designated license.

A copy of the GNU Lesser General Public License -v 1.0 is available at http://opensource.org/licenses/LGPL-3.0

A copy of the Eclipse Public License v 1.0 is available at http://opensource.org/licenses/eclipse-1.0.php

A copy of the Apache Software Foundation License 2.0 is available at http://opensource.org/licenses/apache2.0.php

The following statement must be included in any copies of Toolkit code:

> The S1000D Transformation Toolkit is licensed for use, at the user's election, under the GNU Lesser General Public License, under the Eclipse Public License -v 1.0 or the Apache Software Foundation License v2.0. If, at the time of use, the Project Management Committee has designated another version of these licenses or another license as being applicable to the S1000D Transformation Toolkit, user may select to have its subsequent use of the S1000D Transformation Toolkit governed by such other designated license.
>
> A copy of the GNU Lesser General Public License -v 1.0 is available at http://opensource.org/licenses/LGPL-3.0
>
> A copy of the Eclipse Public License -v 1.0 is available at http://opensource.org/licenses/eclipse-1.0.php
>
> A copy of the Apache Software Foundation License 2.0 is available at http://opensource.org/licenses/apache2.0.php

2. Key Concepts

2.1. S1000D Transformation Toolkit Overview

Conceptually, the Toolkit is a black box that consumes S1000D input files and graphics from a Common Source Database (CSDB) and produces a SCORM content package, mobile web application or PDF output. The input files consist of a SCORM Content Package Module (SCPM) and referenced S1000D data modules.

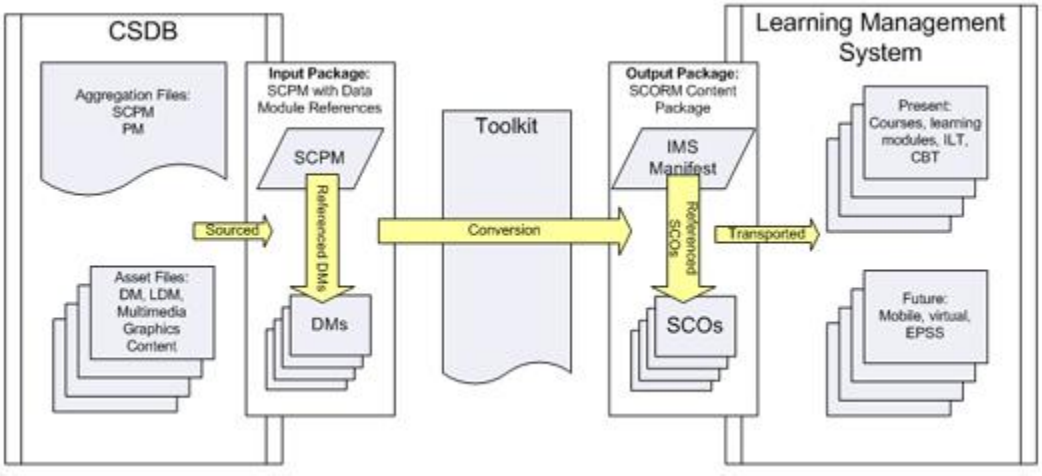

Figure 2.1: Conceptual Model of Toolkit Usage to Produce SCORM Content Package

For SCORM, the Toolkit transforms the SCPM input file into a SCORM IMS Manifest and creates SCOs from the S1000D data modules. These outputs are packaged into the SCORM content package. **Figure 2.1** illustrates SCORM transformations. For the mobile web application, the Toolkit transforms the SCPM and referenced S1000D data modules into html files that can be hosted as a mobile web application. **Figure 2.2** illustrates mobile web application transformations. PDF support will be delivered in a future release.

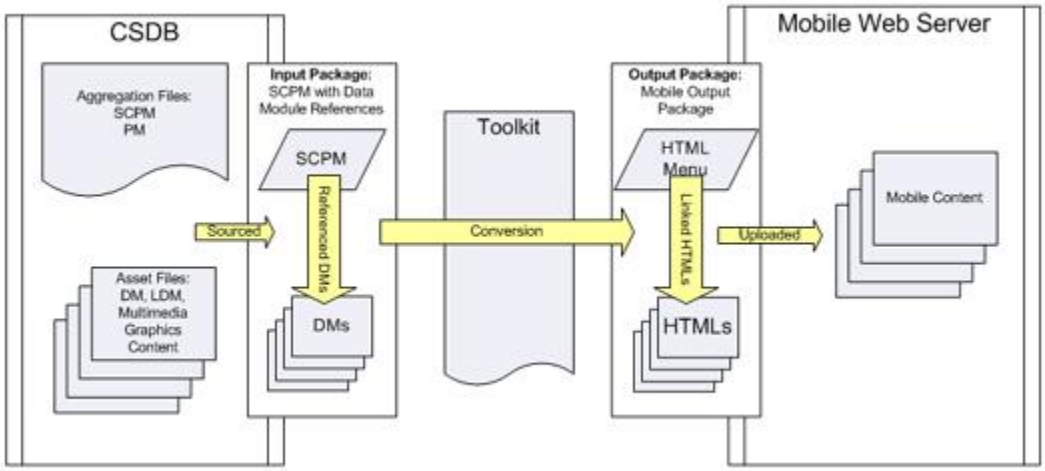

Figure 2.2: Conceptual Model of Toolkit Usage to Produce Mobile Web Output

2.2. Toolkit Building Blocks

2.2.1. Common Source Database (CSDB)

The CSDB is a collection of data modules documenting systems and components. According to the S1000D specification, the CSDB is an information store and management tool for all objects required to produce the technical publications and courseware within projects. The CSDB is the source for the inputs to the Toolkit.

See also:

- http://www.s1000d.org

2.2.2. SCORM Content Package Module (SCPM)

The SCORM Content Package Module (SCPM) is an aggregation file that references S1000D data modules into authoritative course and SCO structures. The S1000D specification defines the SCPM in Chapter 3.2 and Chapter 4.9.5.

The developer organizes the course in the SCPM by referencing S1000D data modules into collections. Each collection is the basis for a SCO. The SCPM and the referenced data modules are the inputs to the Toolkit.

See also:

- http://www.s1000d.org

2.2.3. Data Module (DM)

Data Modules are configurable, reusable pieces of technical information stored in a CSDB. They are uniquely identified by a Data Module Code (DMC). Each data module contains a metadata section called the Identification and Status section. It identifies and describes the data module. A separate Content section contains the technical content. The S1000D specification defines the Data Module in Chapter 3.2.

The developer can organize the course in the SCPM by referencing data modules into the collection.

See also:

- http://www.s1000d.org

2.2.4. Input Package

The Toolkit input package contains the SCPM and referenced S1000D data modules sourced from the CSDB. The input package represents the authoritative source to the output package. The Toolkit output package is a SCORM content package, mobile web application or PDF.

2.2.5. SCORM Output Package

This SCORM output package is a zip file containing the SCORM content package. The content package contains an XML file called an IMS Manifest file, content called SCOs and any required schemas. It can transport the training content to any Learning Management System that is certified to be compatible with the SCORM standard. In the Learning Management System environment, learners can launch and play each SCO, interact with its content and take assessments. The Learning Management System can track overall user course and SCO statuses and provide management reports.

To create the IMS Manifest, XSLT transforms the input package to the output package by mapping the SCPM structures to IMS Manifest structures. To create the physical SCOs, the Toolkit consumes the referenced S1000D data modules and creates them according to the authoritative SCPM organization. Section 5 describes these transformation processes in more detail.

IMS Manifest

The IMS Manifest is an XML document that describes the content structure and associated resources of the SCORM content package. The IMS Manifest contains the transformed elements of the SCPM in a format that an LMS can use.

The IMS Manifest and SCPM have similarities and differences and should be distinguished. The SCPM is an S1000D document within the CSDB that identifies the S1000D data modules, metadata and organizational structure that will be used in the various transformations. While creation of the SCPM is a needed first step to repurpose technical content into SCORM conformant content, it is not an object that a SCORM course can use or one that a Learning Management System (LMS) can understand or process. The LMS requires the IMS Manifest to describe all the metadata, organizations and resources of the SCORM course. It is agnostic of the source of the content. The Toolkit bridges the two documents by transforming the authoritative organizational data of the SCPM to the IMS Manifest data structure.

See also:

- http://www.adlnet.gov

Sharable Content Object (SCO)

The IMS Manifest within a SCORM content package references and organizes Shareable Content Objects (SCOs) in the course. Each SCO is a collection of reusable assets. Each SCO is physically represented by a single launchable learning resource. The SCO is the lowest level of granularity tracked by a Learning Management System. A Learning Management System that contains a SCORM conformant Run Time Environment can launch each SCO and communicate with it through standardized protocols defined by SCORM.

When the Toolkit creates SCOs it uses the S1000D data modules referenced by the SCPM for content. The launchable learning resource is an HTML frameset that is used to display the

navigation and content. For navigation, the frameset provides a list of the data module documents. The order depends on the order in which they were originally listed in the SCPM. For content, the data modules are transformed at runtime from data modules to HTML that is viewed within the frameset. When the Toolkit-produced SCO is viewed in the browser, each data module document is displayed as one screen.

See also:

- http://www.adlnet.gov

2.2.6. Mobile Web Application Output Package

The Toolkit creates the web application output package according to the authoritative structure of the SCPM. To create the mobile web application menu page, the Toolkit uses an XSLT to transform the SCPM into a home HTML (index.htm) and then places it within an output directory, as shown in **Figure 2.2.6**. The home HTML contains a menu. For each scoEntry element in the SCPM, a clickable item is created in the menu. Since a scoEntry is a collection of DMs, when a user clicks a menu item at runtime, the web application will display the content of all the DMs, each in its own page, navigated sequentially. The order of DMs displayed to the user is the order in which they are referenced in the SCPM.

To create each individual web application page, the Toolkit transforms each DM into an individual HTML page and places each one into a directory associated with its parent scoEntry, according to the SCPM. The naming convention for a transformed DM is: <data module file name>.htm. Each scoEntry is indexed with a number and can contain one or more DMs. For example, **Figure 2.2.6** shows four transformed DMs placed within a scoEntry directory with an index of 17. The highlighted DM in the figure is:

```
17/ DMC-S1000DBIKE-AAA-D00-00-00-00AA-100A-A-T2GC_001-00_EN-US.xml.htm
```

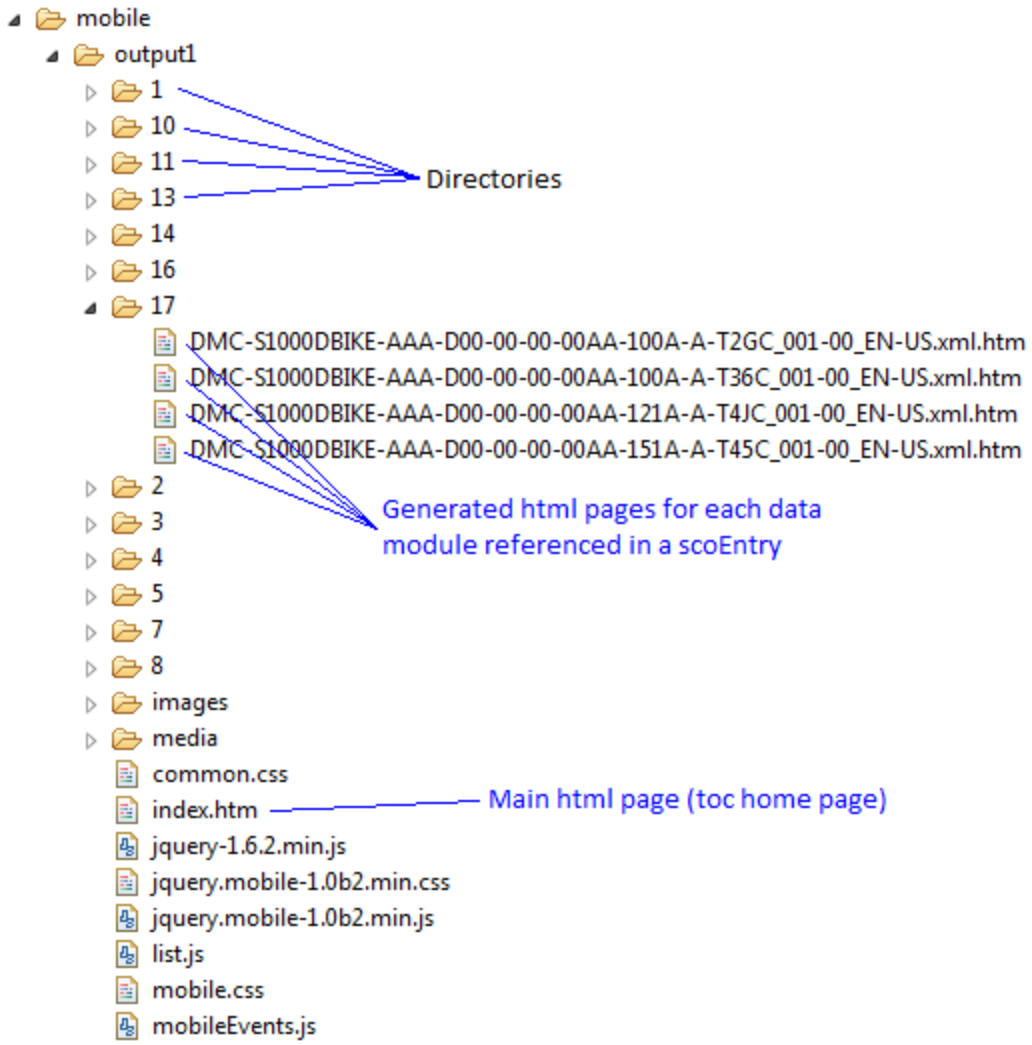

Figure 2.2.6: Web Mobile Output

The transformed menu page and data modules are wrapped in the JQuery Mobile "page" structure that allows for the application of the Jquery Mobile Framework. The mobile web application output package contains all of the generated HTML pages, all referenced S1000D ICN files, the required Jquery Mobile JavaScript and CSS files, plus additional JavaScript and CSS files need for navigation and formatting. This output package can then be hosted on any web server or ported to a native mobile application with the use of various tools (ex. PhoneGap or Appcelerator Titanium).

2.2.7. PDF Output Package

The output of the PDF package is the PDF file containing the content of the S1000D course and the assessments for the course. To view the PDF outputs use Adobe Acrobat to view the output. Adobe Acrobat is available at http://get.adobe.com/reader/otherversions/

3. Getting Started

3.1. System Requirements

The Toolkit is written in Java and requires Java SE 6 (JRE) to be installed to execute the application and the Java code.

3.2. Setup and Installation

The Toolkit package comes with all required tools except the Java JDK and the binary distribution of Apache Ant.

3.2.1. Toolkit Distributions

The Toolkit is currently available in a full source distribution format.

The full source distribution contains the source and executable code for the Toolkit plus all of the documentation describing how to use the Toolkit. This distribution will allow you to modify Toolkit Java code and look at how the Toolkit works.

The distribution is available for download from https://sourceforge.net/projects/s1000d-scorm/

3.2.2. Installation

To install the full package:

- Download the full package from https://sourceforge.net/projects/s1000d-scorm/
- Unzip the package into the C:\ directory on Windows, or into a home directory on Linux.

Note: You cannot run a Toolkit build until you have installed the Java JDK and Apache Ant.

3.2.3. Installing the Java JDK

You should always ensure that previous versions of Java and Ant are uninstalled before installing new ones.

Installing the JDK on Windows:

To obtain the latest version of the Oracle JDK, enter the URL
http://www.oracle.com/technetwork/java/javase/downloads/index.html

1. Locate the "Download JDK" button to download the latest version. (The Toolkit was built using Java SE 6).
2. Select the appropriate platform and language for your system. (Accept the License Agreements and complete all necessary forms on the site)
3. Save and install the .exe file.
4. Set the JAVA_HOME environment variable to location where you installed JDK.
5. Add the bin directory to your PATH environment variable.

For more details on setting environment variables, reference this section: Setting Environment Variables on Windows.

3.2.4. Installing the Apache Ant

You should always ensure that previous versions of Java and Ant are uninstalled before installing new ones.

Installing Ant on Windows:

1. To obtain the latest version of Apache Ant, enter the URL: http://ant.apache.org/bindownload.cgi.
2. On the Apache Ant Project page, find the heading, Current Release of Ant.
3. Select apache-ant-1.8.2-bin.zip. Note that if you need to download encrypted versions, there are other links: PGP, SHA1, SHA512 and MD5.
4. Click Save to save it to your C:\ directory. Unzip it to be in the root of C, for example: C:\apache-ant-1.8.2. As a second check, ensure there is a bin directory within the ant directory: C:\apache-ant-1.8.2\bin.
5. Set the ANT_HOME environment variable to location where you saved Ant. Ex. C:\ apache-ant-1.8.2.
6. Add the bin directory to your PATH environment variable.

For more details on setting environment variables, reference this section: Setting Environment Variables on Windows.

3.2.5. Setting Environment Variables on Windows

The installation steps for Java and Ant require the setting of home and path variables. Start by accessing the Environment Variables form:

1. From the Start Menu, select Start > Settings > Control Panel.
2. Double-click System to open the System Properties window.
3. On the Advanced tab, click the Environment Variables button.

For the Java and Ant Home environment variables:

1. Inspect the System Variables table for JAVA_HOME.
2. If the JAVA_HOME variable already exists, then edit it. Otherwise create a new variable called JAVA_HOME.
3. Set the variable to the location where you installed Java. For example: C:\Program Files\Java\jdk1.6.0_21.
4. Inspect the System Variables table for ANT_HOME.
5. If the ANT_HOME variable already exists, then edit it. Otherwise create a new variable called ANT_HOME.
6. Set the variable to the location where you saved Ant. For example: C:\apache-ant-1.8.2.

For the PATH environment variable:

1. If the Path variable already exists, then edit it. Otherwise create a new variable called Path.

2. Add this text to the Path variable: %ANT_HOME%\bin;%JAVA_HOME%\bin. Note that all paths must be separated by a semicolon.

3.2.6. Contents of the Toolkit

Figure 3.2.6 shows the directories and files of the Toolkit. The table following describes each directory.

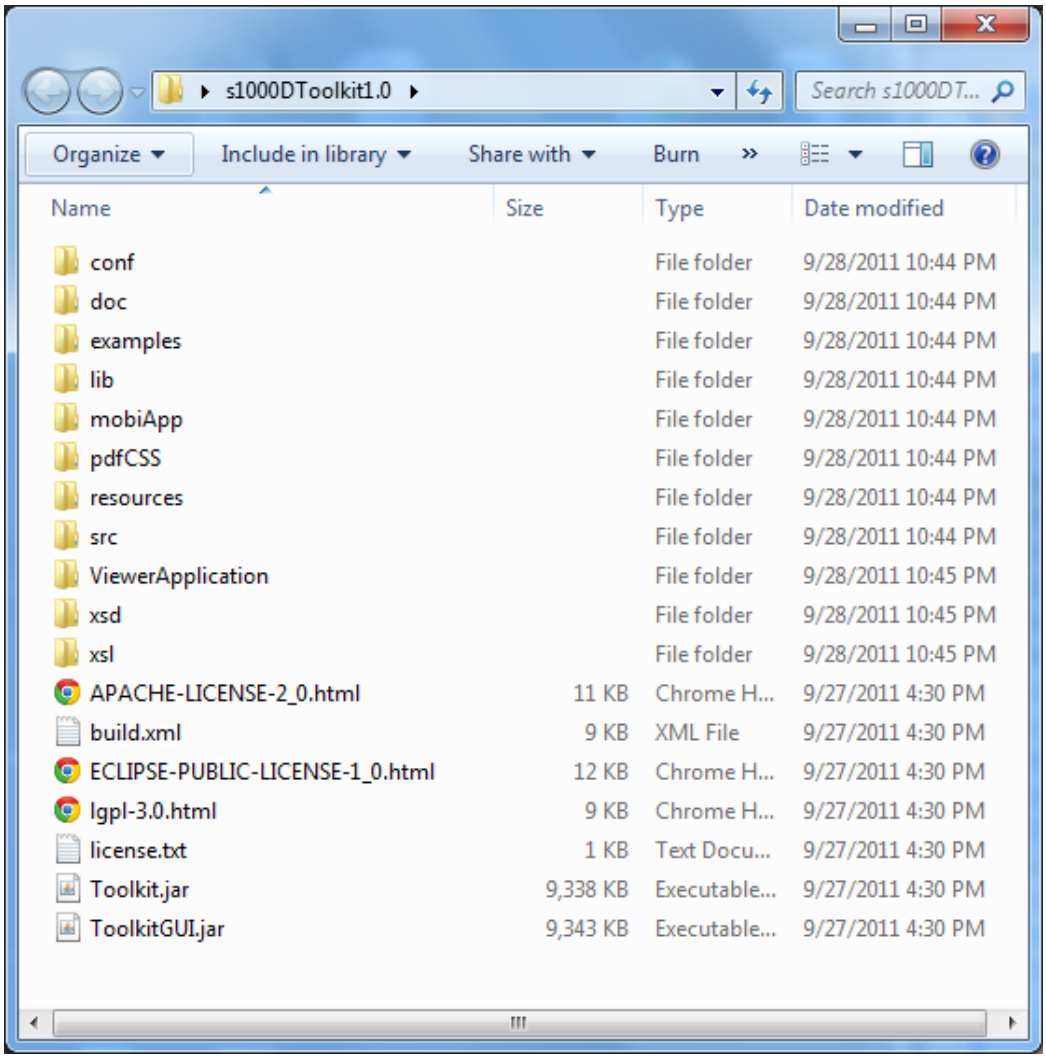

Figure 3.2.6: Toolkit Directories and Files

Directory	Description
root	The root directory contains the Ant build.xml file, license files such as APACHE-LICENSE-2.0.html, ECLIPSE-PUBLIC_LICENSE-1_0.html and license.txt. The executable jar files Toolkit.jar (command line executable) and ToolkitGUI.jar (Java GUI executable).

	This GUI was implemented to give the developer a feel for the functionality of the Toolkit and is not designed to be integrated within a vendor application.
conf	The conf directory contains the Apache Commons Chain Catalog XML document.
doc	The doc directory contains the JavaDoc files.
examples	The example directory contains the S1000D example bike data. Examples exist for both 4.0 and 4.1 bike data. The bike_resource_package directory contains the data module resource XML files for the Bike demonstration project. It also contains all image files referenced by the resource XML files. The bike_SCPM directory contains the S1000D SCPM for the Bike demonstration project.
lib	The lib directory contains libraries required by the executable. The directory includes jars for the Apache Commons Chain. It also includes jars for parsing, manipulating and outputting XML.
logs	The logs directory contains output logs for Toolkit executions.
mobiApp	The mobiApp directory contains CSS and JavaScript files needed for the mobile web application output.
pdfCSS	The pdfCSS directory contains the CSS files needed for the PDF output.
mobile	The mobile directory contains the mobile web application output content created by the Toolkit.
resources	The resources directory contains miscellaneous resource file.
src	The src directory contains the Java source code.
ViewerApplication	The ViewerApplication directory contains the .xslt, .css and other files needed to render the S1000D data modules in a browser.
xsd	The xsd directory contains the ADL and LOM XML Schema files.
xsl	The xsl directory contains the stylesheet that converts the SCPM to an imsmanifest.xml file.

3.3. Try the Toolkit

A Toolkit sample is provided to convert SCPM and resource files for the Bike sample project into the SCORM content package, mobile web application or PDF format. A GUI interface is provided to demonstrate this functionality. This is not meant to be used in a production environment. Here are the steps to run the GUI and create the desired output.

Step 1: Locate the jar file, ToolkitGUI.jar, in the Toolkit download. Double-click the ToolkitGUI.jar to reveal the GUI. For this example, the output selection is SCORM Content Package. For other outputs, make a different selection. See **Figure 3.3a**.

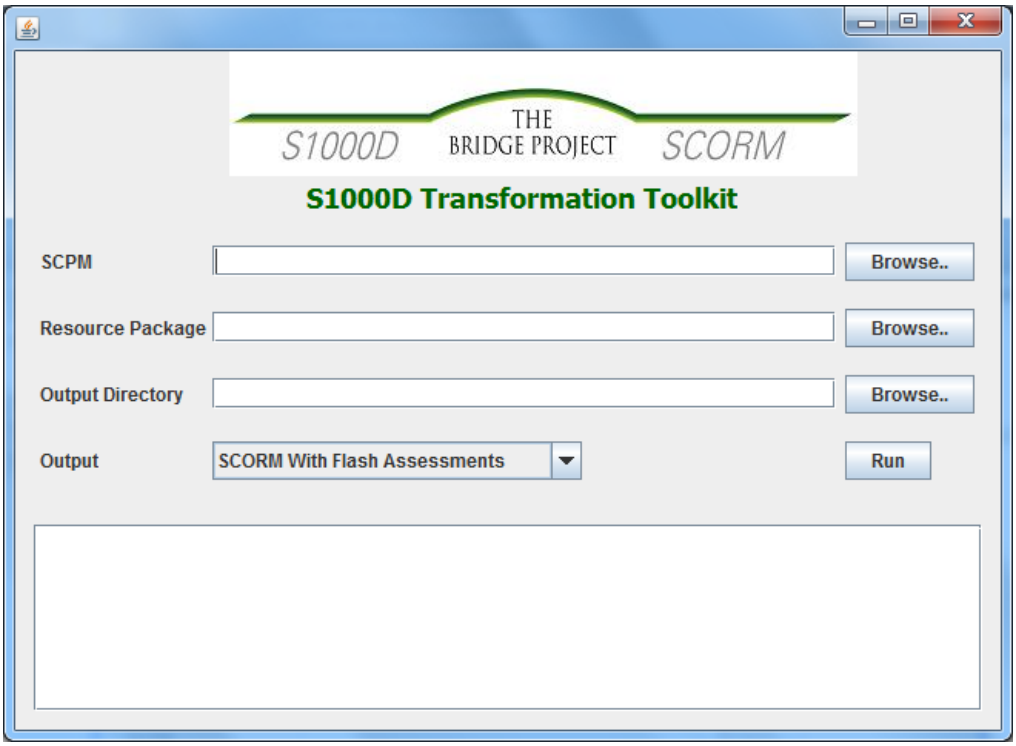

Figure 3.3a: Toolkit GUI

Step 2: Click the browse button to search for the SCPM. Select the SCPM XML file and click Open. **Figure 3.3b** shows the selected sample Bike project SCPM. This Toolkit download contains a bike_SCPM directory for both S1000D Release 4.0 and 4.1.

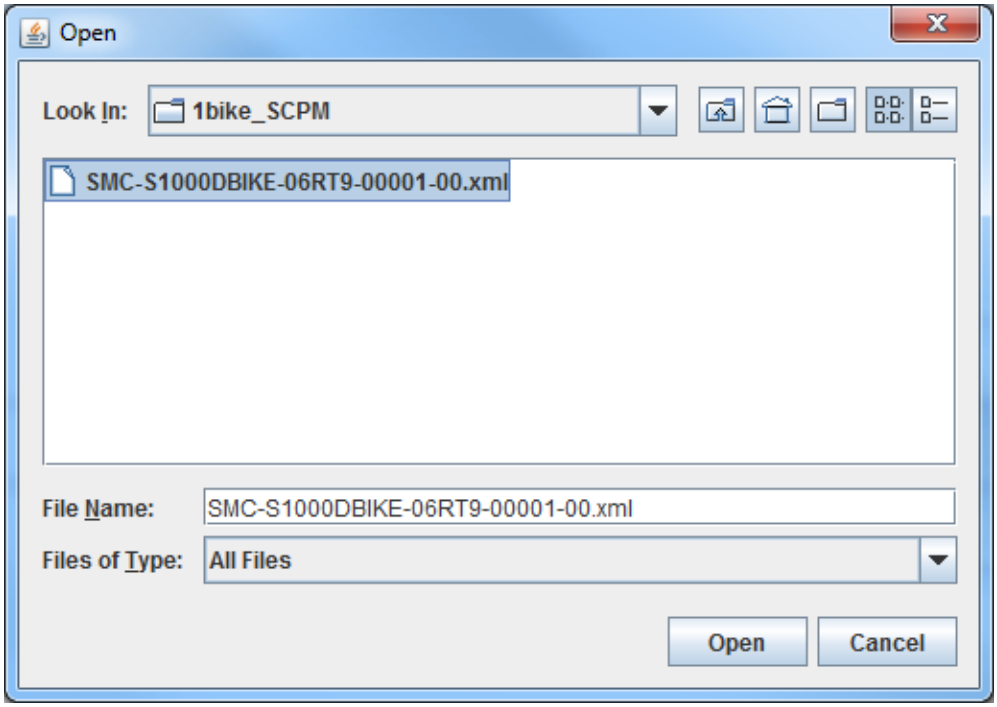

Figure 3.3b: Toolkit Search for SCPM

Step 3: After clicking the Open button, you will be returned to the Toolkit GUI, click the browse button to search for the Resource Package. Select the Resource Package and click Open. **Figure 3.3c** shows the selected sample Bike project Resource Package. This Toolkit download contains a bike_resource_package for both S1000D Release 4.0 and 4.1.

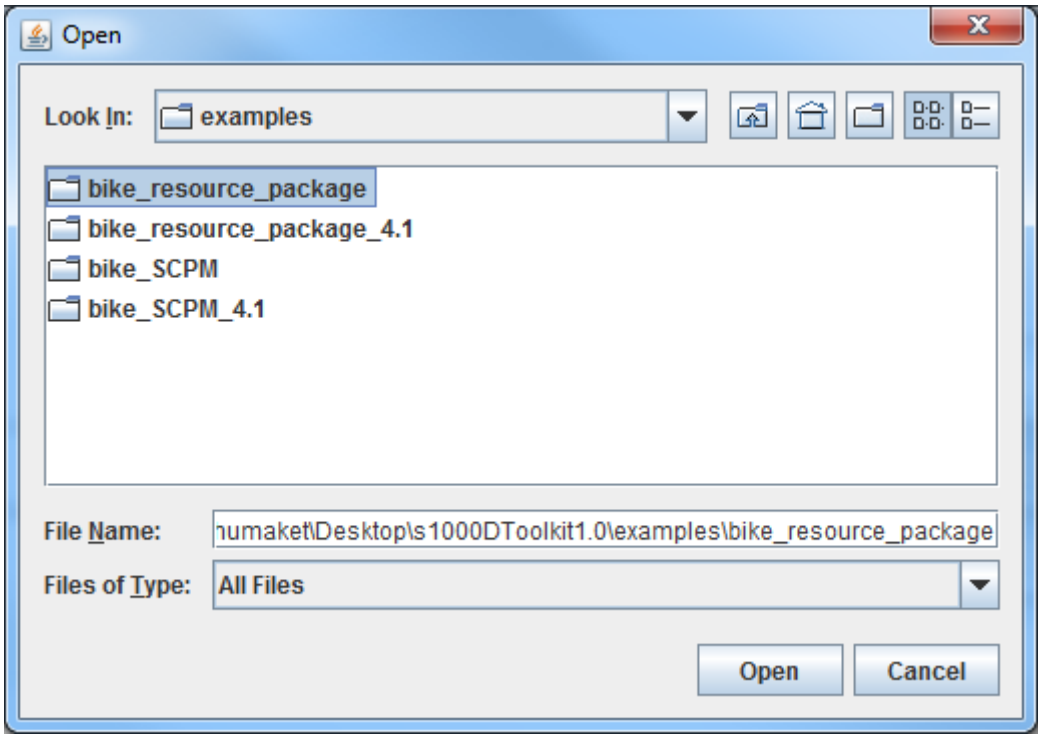

Figure 3.3c: Toolkit Search for Resource Package

Step 4: After clicking the Open button, you will be returned to the Toolkit GUI, click the browse button to search for the Output Directory. Select an output directory and click Open. The output of the toolkit will be written to this directory.

Step 5: After the SCPM and Resource Package are selected, you will be returned to the Toolkit GUI. Now click Run. The Toolkit will create the IMS Manifest and the content package. **Figure 3.3d** shows the results of the successful run.

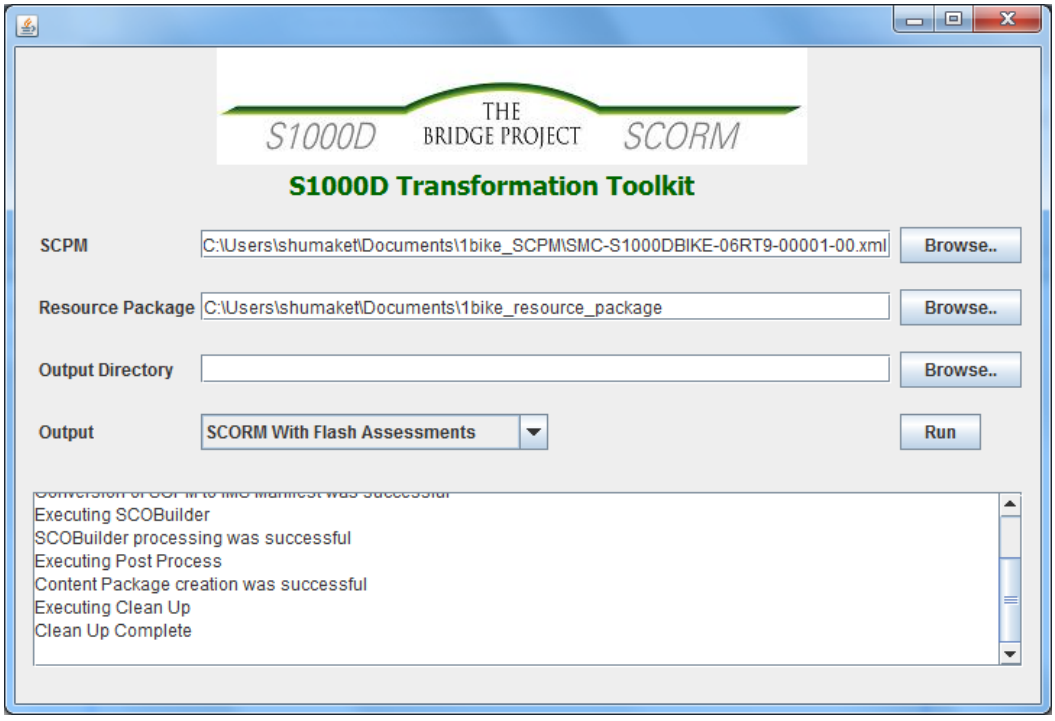

Figure 3.3d: Toolkit Successful Run

The Toolkit has now created a logs directory and selected output format. The SCORM output is a SCORM zip file. For the Bike Project that will be S1000D_BIKE_Learning_Sample.zip. The mobile web application output is in the mobile directory. These directories and files are found in the Toolkit directory if no output directory is specified.

3.4. How Developers Will User the Toolkit

CSDB Vendors, LMS Vendors and Application Developers can use the Toolkit in three scenarios:

- Publishing SCORM from an S1000D Compliant Application
- Importing S1000D Content into a Learning Management System
- Independent Application Development

3.4.1. Publishing SCORM from an S1000D Compliant Application

In **Figure 3.4.1**, CSDB vendors produce S1000D applications that publish content that conforms to the S1000D standard. CSDB developers will be able to integrate this Toolkit into their code so that their product will be able to publish SCORM content package, mobile web application and PDF outputs.

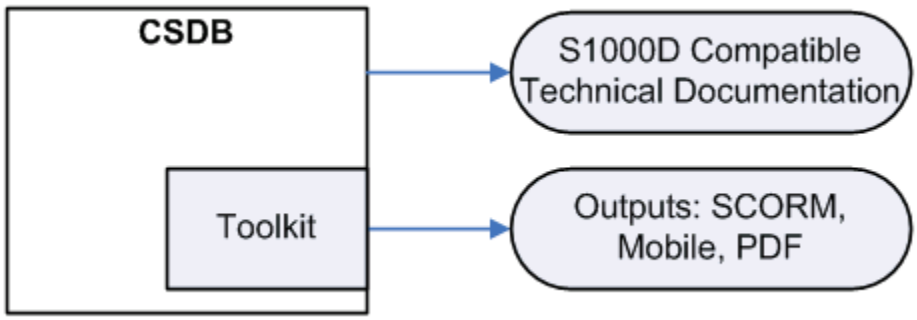

Figure 3.4.1: Publishing from CSDB

3.4.2. Importing S1000D Content into a Learning Management System

In **Figure 3.4.2**, LMS vendors produce applications that courseware developers use to assemble training content and publish in various formats, including SCORM, mobile web application and PDF. LMS vendors will be able to integrate this Toolkit into their code so that their product will be able to import S1000D content, and publish as one of those outputs. In this usage, training developers will be able to import the S1000D content and share, duplicate or modify it to conform to training design standards, before publishing it.

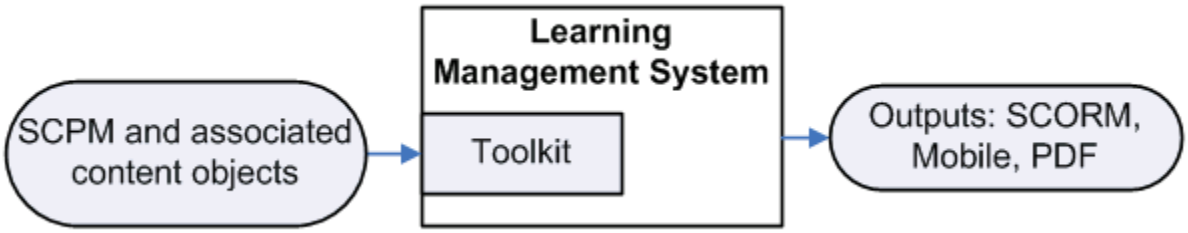

Figure 3.4.2: Importing S1000D Content into a Learning Management System

3.4.3. Independent Application Development

In **Figure 3.4.3**, an independent application can be created that will take the SCPM and associated resource package, from any S1000D application, as inputs, and output a SCORM content package, mobile web application or PDF output. The SCORM content package can be

played on any Learning Management System that is certified to be conformant with the SCORM standard. The mobile web application can up uploaded to any web server. The PDF can be widely distributed.

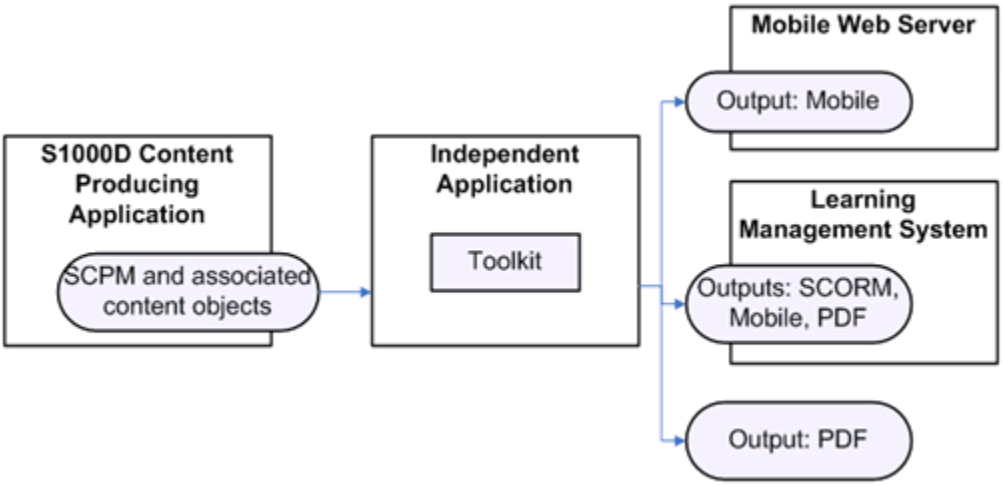

Figure 3.4.3: Independent Application Development

4. Using the Toolkit

4.1. The Input Package

The Toolkit requires two inputs:

- The SCPM: The SCPM is an XML file that is an output of S1000D. For the Bike demonstration project it is found in the directory, examples/bike_SCPM.
- The Resource Package directory: This directory contains the data module resource XML files referenced by the SCPM. For the Bike demonstration project it is the examples/bike_resource_package directory.

4.2. Toolkit Local Files Needed for SCORM

The Toolkit contains two directories that are used to produce a SCORM content package: ViewerApplication and xsd.

4.2.1. Viewer Application

Figure 4.2.1 shows all the files required to render the data modules in the browser and to do the SCORM API communication at runtime. Examples include APIWrapper.js, s1000d_4.xslt, navScript.js and SCO header images.

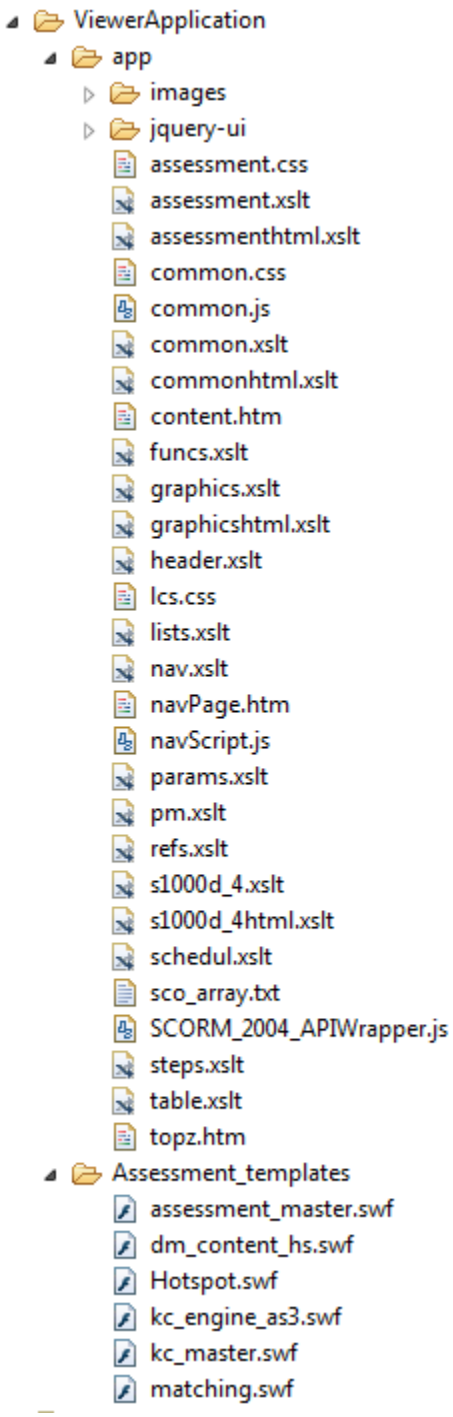

Figure 4.2.1: Viewer Application Files

4.2.2. XSD

Figure 4.2.2 shows all of the IEEE Learning Object Metadata (LOM) schemas files, the ADL SCORM schema files and IMS content package schema version IMS CP 1.1.4 (ADL and IMS schemas are highlighted).

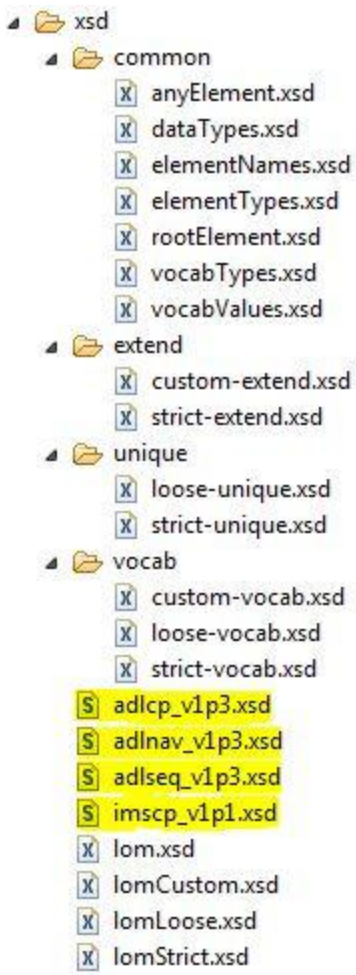

Figure 4.2.2: Schema Files

4.3. Toolkit Local Files Needed for Mobile Web Application

The Toolkit contains one directory that is required for the mobile web application output, mobiApp.

4.3.1. mobiApp

Figure 4.3.1 shows all the files required to transform the SCPM and Data Modules into the mobile web application output.

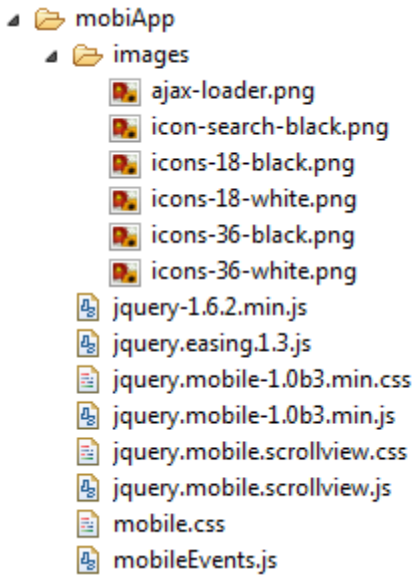

Figure 4.3.1: Mobile Web Application Files

4.4. Running the Toolkit, Command Line SCORM Transformation

The Toolkit is also implemented into an application using command line interface. These are the instructions for running the Toolkit from the command line:

1. Click Start > Run from the toolbar.
2. Type cmd in the Open field.
3. Change the command prompt to the Toolkit location, e.g., c:\s1000DToolkit1.0.
4. Submit the command for SCORM, mobile web application or PDF, as provided in the following subsections.

Confirmation messages will be displayed when the Toolkit processing completes successfully.

4.4.1. SCORM Command

Here is the command syntax for the SCORM output: `java –jar Toolkit.jar "<scpm location>" "<resource package directory>" <–scormflash | –scormhtml> "<output directory>"`

SCORM examples:

```
java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\s1000DToolkit1.0\examples\bike_resource_package" –scormflash
```

```
java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\s1000DToolkit1.0\examples\bike_resource_package" –scormhtml "C:\myFolder"
```

4.4.2. Mobile Web Application Command

Here is the command syntax for the mobile web application output: `java –jar Toolkit.jar "<scpm location>" "<resource package directory>" <-mobileperformancesupport | - mobilecourse>  "<output directory>"`

Mobile web application examples:

```
java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\s1000DToolkit1.0\examples\bike_resource_package" -mobileperformancesupport
```

```
java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\s1000DToolkit1.0\examples\bike_resource_package" -mobilecourse "C:\myFolder"
```

4.4.3. PDF Command

Here is the command syntax for PDF output: `java –jar Toolkit.jar "<scpm location>" "resource package directory"> <-pdfstudent | -pdfinstructor> "<output directory>"`

PDF examples:

```
java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\s1000DToolkit1.0\examples\bike_resource_package" -pdfstudent
```

```
 java -jar Toolkit.jar "C:\s1000DToolkit1.0\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml"
"C:\s1000DToolkit1.0\examples\bike_resource_package" -pdfinstructor
"C:\myFolder"
```

4.4.4. Toolkit Successful Completion

When the Toolkit processing completes successfully, the confirmation messages displayed depends on the output selected.

For SCORM:

- Executing S1000D Converter
- Conversion of SCPM 4.1 to SCPM 4.0 was successful
- Executing PreProcess
- Conversion of SCPM to IMS Manifest was successful
- Executing SCOBuilder
- SCOBuilder processing was successful
- Executing Post Process
- Content Package creation was successful
- Executing Clean Up
- Clean Up Complete

For mobile web application:

- Executing S1000D Converter
- Conversion of SCPM 4.1 to SCPM 4.0 was successful
- Executing Mobile Builder
- MobileBuilder processing was successful
- Executing Clean Up
- Clean Up Complete

For PDF:

- Executing S1000D Converter
- Conversion of SCPM 4.1 to SCPM 4.0 was successful
- Executing PDF Builder
- Successfully created PDF
- Executing Clean Up
- Clean Up Complete

4.5. SCORM Transformation

The Toolkit's SCORM transformation process converts the content aggregation structure defined in the SCPM to the aggregation structure of a SCORM IMS Manifest file. All titles, metadata and referenced files from the SCPM are mapped to specific XML structures that define the IMS Manifest file.

The Toolkit locates the SCPM and resource folder by way of the user provided URL parameters. The SCPM and resource folder URLs provide the locations of all the files needed to transform the S1000D managed content referenced in the SCPM to a SCORM content package. The Toolkit locates and gathers the content files into the folder structures referenced within the IMS Manifest. The content is also grouped within the IMS Manifest into SCO organizations to reflect the collection of SCOs defined in the SCPM.

The Toolkit transformation process also provides a Viewer Application (viewer) for the output SCORM content package. The viewer is a browser-based rendering environment. It consists of a collection XML style sheet translation (XSLT) files along with other files necessary to provide content navigation and other features required for presentation of the content in a Web browser such as Internet Explorer. The viewer is required because the S1000D content is left in its original XML form. The Toolkit adds XML processing instructions to the S1000D XML content files to enable the viewer rendering operations.

The generated SCO files, all of the Viewer Application files and all of the necessary SCORM schema files are then added as references in the imsmanifest.xml structures to complete the transformation. The imsmanifest.xml file and all of the required resources (S1000D data

modules, S1000D ICN files, SCO files, Viewer Application files and the SCORM schemas) are then packaged into a .zip file to produce the output SCORM content package.

NOTE: Currently the Toolkit does not perform validation on the content package. If there is a reference to a missing data module or ICN file in the SCPM or any of the referenced data modules then the imsmanifest.xml file produced by the Toolkit could result in SCORM content package validation errors.

4.6. Reviewing the SCORM Output Package

The Toolkit consumes the inputs, processes the transformation procedures and produces a SCORM content package. It is placed inside the "s1000DToolkit1.0" directory. The name of the content package will be:

- <value of the scormContentPackageTitle element from the SCPM with _ replacing any whitespaces>.zip
- Example: S1000D_BIKE_Learning_Sample.zip

NOTE: If the provided Toolkit GUI is used to run the Toolkit then the confirmation messages or error messages will be saved in the directory "s1000DToolkit1.0/logs" to files named toolkit-<time stamp>.log. This feature is not currently available when the Toolkit.jar file is used to run the Toolkit.

When the Bike sample course is uploaded to a SCORM Conformant LMS and run, it will display the Bike course. **Figure 4.5** is a screenshot of one of the content screens.

Figure 4.5: Bike Sample Course

5. Integrating the Toolkit

5.1. Conceptual Architecture Modules

The conceptual model in **Figure 5.1** is broken down into modules and sub modules that will allow for expandability and allow for plug-in support for future desired formats.

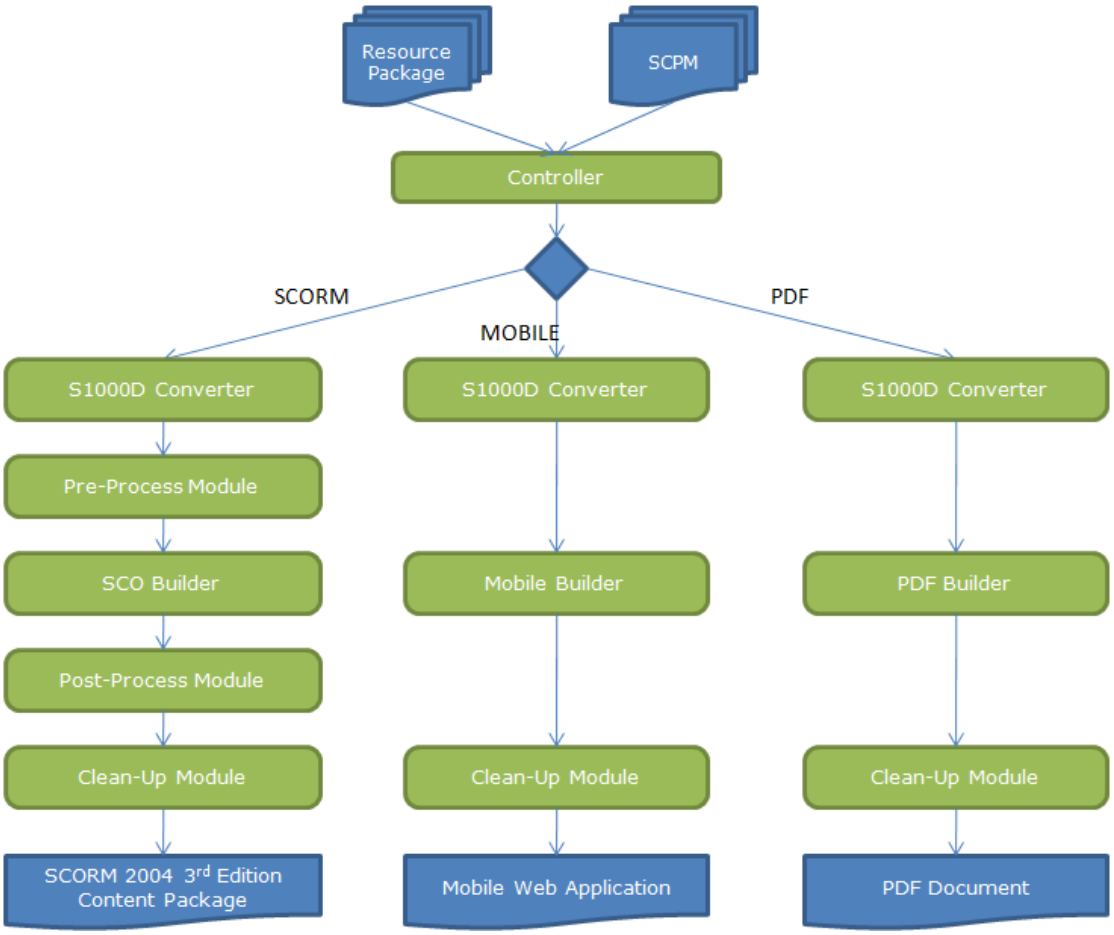

Figure 5.1: Toolkit Conceptual Architecture

5.2. Apache Commons Chain

The Apache Commons Chain supports the Toolkit's conceptual Architecture by providing an API that facilitates the "Chain of Responsibility" design pattern. For more information see: http://commons.apache.org/chain/

Each module in the Toolkit will be "command" that has an execute() method. Each execute() method is passed a "context" parameter that isolates the command implementations from the environment. The Toolkit will utilize the "context" to manage the input required between modules. These are the default input key-value pairs that will be required for the Toolkit.

SCPM_FILE - String that represents the location of the S1000D SCPM file.

RESOURCE_PACKAGE - String that represents the location of the resource package that contains all the S1000D files referenced in the SCPM.

XML_SOURCE - The IMS Manifest file that is generated from the SCPM.

CP_PACKAGE - File that represents the directory that is being used to build the SCORM content package.

URN_MAP – File that provides a way to map the URN values for the resources to the file names.

PDF_OUTPUT_OPTION - String that specifies if the output of PDF should be the instructor PDF or the student PDF. Values are -student of –instructor

OUTPUT_DIRECTORY - String containing the output directory of the toolkit.

MOBLIE_FILES_TO_DELETE - An ArrayList containing the paths to the files that need to be deleted after a successful run of the mobile output.

5.3. Controller

The Controller class is the initial start of the Toolkit where the parameters (SCMP, resource package and desired output) are passed into. Based off of the desired output the controller initialized the appropriate chain of commands. These chains are controlled in the chain-config.xml file. This configuration is further explained in section 6.4, Modifying the Apache Commons Configuration.

5.4. SCORM Chain

5.4.1. S1000D Conversion Module

Input: SCPM_FILE (Version 4.1), RESOURCE_PACKAGE

Output: SCPM_FILE (Version 4.0)

Description: Converts S1000D 4.1 learning data into S1000D 4.0 learning data so that it can be processed by the toolkit. If /scormContentPackage/content/scoEntry/scoEntryItem exist is a 4.1 SCPM the SCPM file will be converted to a 4.0 SCPM.

5.4.2. Pre-Process Module

Input: SCPM_FILE, RESOURCE_PACKAGE

Output: XML_SOURCE (The IMS Manifest file but will not have all the necessary information at this point to be SCORM conformant.)

Description: The preprocessing is where the mapping from SCPM to SCORM IMS Manifest file begins (XSLT transform). The reason for using a content aggregation structure that is a

standard is important. It is so that the pre process can be easily swapped out to support other inputs in the future without affecting the rest of the Toolkit.

The first step takes all of the data module files and media files (ICN) in the RESOURCE_PACKAGE to create a URN Resource Map, see **Figure 5.4.1a**. Each ICN referenced by a data module is identified and located based on the <!ENTITY> elements in that data module. Another reason for the URN_MAP to be produced is so that the "resource/dependency" elements in the XML_SOURCE can be built in a later module.

Figure 5.4.1a: Resource URN Map

The URN Resource Map is structured in this way:

```
<urn name="URN:S1000D:DMC-S1000DBIKE-AAA-D00-00-00-00AA-932A-T-H30A">
  <target type="figure">DMC-S1000DBIKE-AAA-D00-00-00-00AA-932A-T-A_001-00_en-
us.xml</target>
</urn>
```

Next the SCPM_FILE is transformed with an XSLT file to create the XML_SOURCE, see **Figure 5.4.1b.**

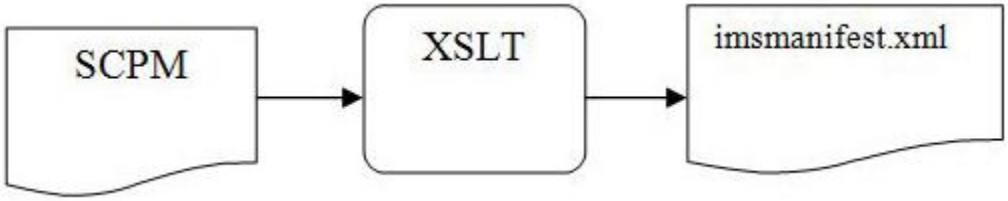

Figure 5.4.1b: Transformation

Then all of the files from the URN Resource Map are added to the XML_SOURCE as "resource" elements. The output of the Pre-Process module is the Manifest file. **Figure 5.4.1c** shows the mapping of elements in the final manifest file to source elements in the original SCPM.

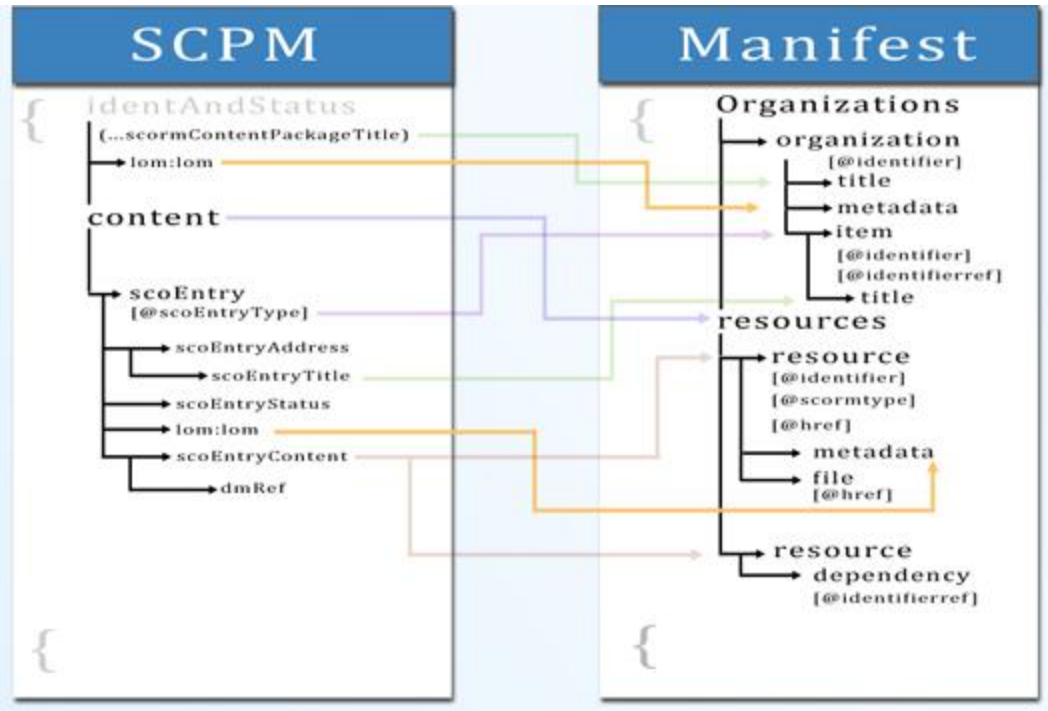

Figure 5.4.1c: SCPM - IMSMANIFEST Mapping

5.4.3. SCO Builder

Input: XML_SOURCE, CP_PACKAGE

Output: HTML SCOs, updated XML_SOURCE

Description: This is where the launchable learning resources are created. The default output of the Toolkit is HTML SCOs. The CP_PACKAGE is created during this module. The files from the resource package are copied over to the CP_PACKAGE and placed in a directory named "resources/s1000d". The Toolkit Viewer Application files (from Toolkit Local Files) are also copied over to the "resources/s1000d" directory. The created HTML SCOs will be copied over to the directory named "resources/scos.

The XML_SOURCE (IMS Manifest file) will be updated during this process. See **Figure 5.4.2**: SCO Builder. The location of the HTML SCOs will be used as the "href" attributes on the appropriate "resource" elements. The Viewer Application files will be added as a common asset "resource" element and a "dependency" element referencing this common resource will be added to all the "resource" elements that contain SCOs.

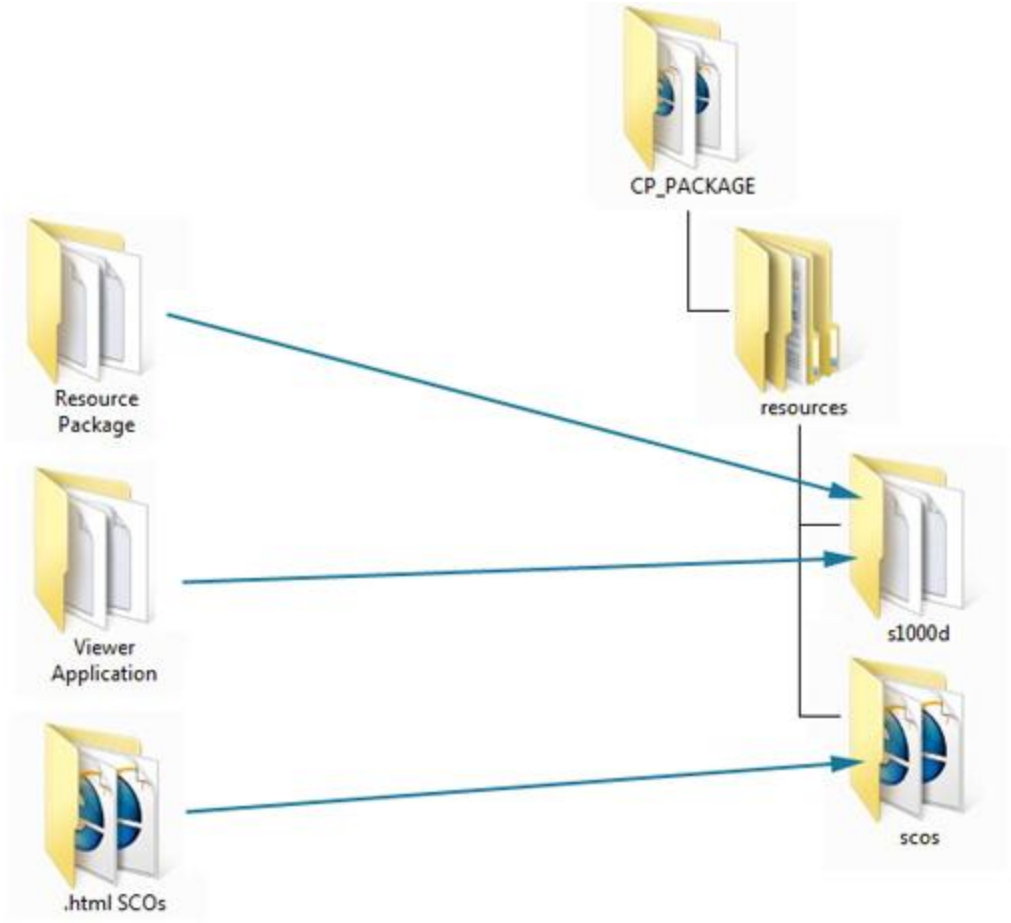

Figure 5.4: SCO Builder

5.4.4. Post-Process Module

Input: XML_SOURCE, CP_PACKAGE

Output: SCORM content package

Description: The post process will copy the ADL Schemas, IEEE LOM Schemas and XML_SOURCE to the CP_PACKAGE and finally zip the file up into a SCORM content package. Any cleanup of the Toolkit can happen here as well. See **Figure 5.5:** Post Process.

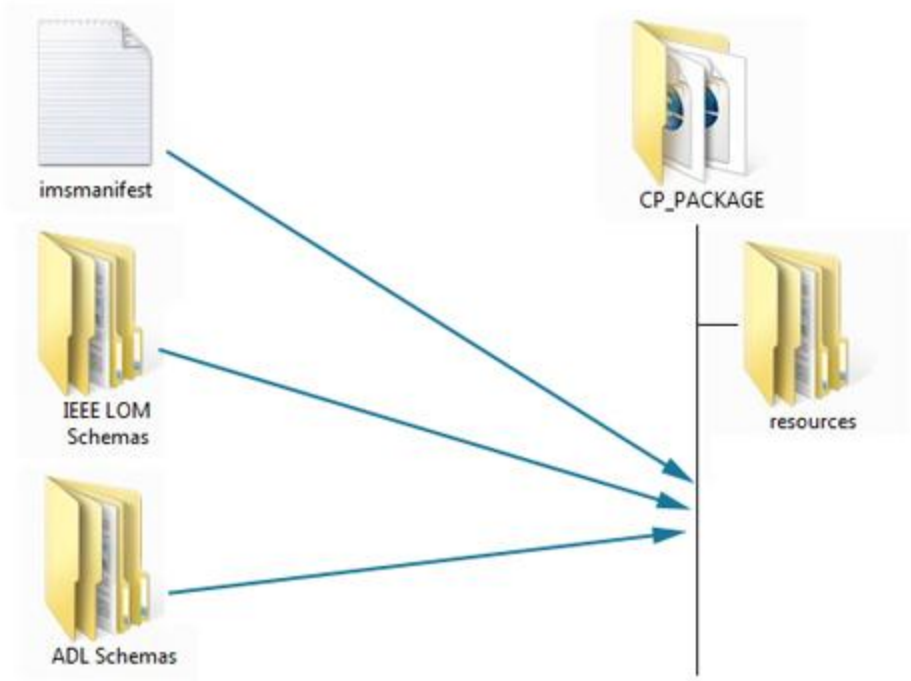

Figure 5.5: Post Process

5.4.5. Clean Up Module

Input: None

Output: None

Description: Resets the context object so that values that may cause errors in subsequent runs will be removed. Any folders that may have been created by the modules of the Toolkit are also deleted.

5.4.6. Viewer Application

The Toolkit Viewer application is a rendering engine packaged with the toolkit SCORM output. The viewer was developed to serve as a rendering engine for the S1000D Bike sample for S1000D learning content. While it provides support for a broad range of S1000D data elements, it is not intended to serve as a comprehensive rendering engine for the entire spectrum S1000D

data modules or data elements. It is provided only to serve as an example of how S1000D learning content can be rendered for display in a Web browser.

The viewer is a Web browser-based rendering environment. It consists of a collection XML Style sheet translation (XSLT) files along with other files necessary to provide content navigation and other features required for presentation of the content in a Web browser such as Internet Explorer. The viewer is required because the S1000D content is left in its original XML form. The Toolkit adds XML processing instructions to the S1000D XML content files to enable the viewer rendering operations.

The viewer application also supports a basic implementation of the ADL/SCORM runtime environment (RTE). Viewer support for the RTE is extended to the SCO initialize and SCO terminate API methods. The viewer navigational and SCORM API scripting are contained in the navScript.js file.

Users are free to modify the viewer application transformations and application scripting under the conditions of the Toolkit copyright.

5.5. Mobile Chain

The S1000D Conversion Module and Clean Up Module are run in the mobile chain

5.5.1. Mobile Builder Module

Input: SCPM_FILE, RESOURCE_PACKAGE, OUTPUT_DIRECTORY

Output: Mobile Web Application

Description: The web application builder is where the mobile web application files are created. The SCPM_FILE and all of the files in the RESOURCE_PACKAGE are transformed into HTML files that are wrapped with the jQuery Mobile Framework.

1. The first step takes all of the data module files and media files (ICN) in the RESOURCE_PACKAGE to create a URN Resource Map. The URN Resource Map is used in the transformation of the data modules.
2. The SCPM_FILE is then transformed to the home HTML page (index.htm) using xsl/bridge/toolkit/commands/scpmStylesheet.mobile.xsl.
3. Each data module referenced by the SCPM_FILE is then transformed into individual HTML pages to provide the content using xsl/bridge/toolkit/commands/dmStylesheet.mobile.xsl. This XSLT file utilizes some of the XSLT files from the SCORM Viewer Application.
4. A list.js file is then generated to control the navigation between pages in the mobile web application.
5. Finally all of the jQuery Mobile files plus additional required CSS and JavaScript files are copied over to the mobile output directory.

5.6. PDF Chain

The S1000D Conversion Module and Clean Up Module are run in the PDF chain.

5.6.1. PDF Builder Module

Input: SCPM_FILE, RESOURCE_PACKAGE, PDF_OUTPUT_OPTION, OUTPUT_DIRECTORY

Output: PDF file

Description: The PDF Builder Module is the command that creates the PDF output of the toolkit. The SCPM_FILE and all of the files in the RESOURCE_PACKAGE are transformed into the PDF file.

6. Modifying the Toolkit Output

The toolkit is open source and can be modified to publish variants of the output formats. This section provides instructions for the most common modifications. After making changes, you will need to create new jar files for the Toolkit. Then you will republish the output using the new jar files you created.

6.1. Modifying the SCORM Output

The content package contains many graphics that serve as interface elements. These include header, footer and navigation graphics. The process to modify one of these graphics so that an output course is published with a different look is very easy.

You need to know which graphic(s) you need to change. **Figure 6.1** shows the interface elements for the header and footer in the output course. The following table provides their filenames and purpose.

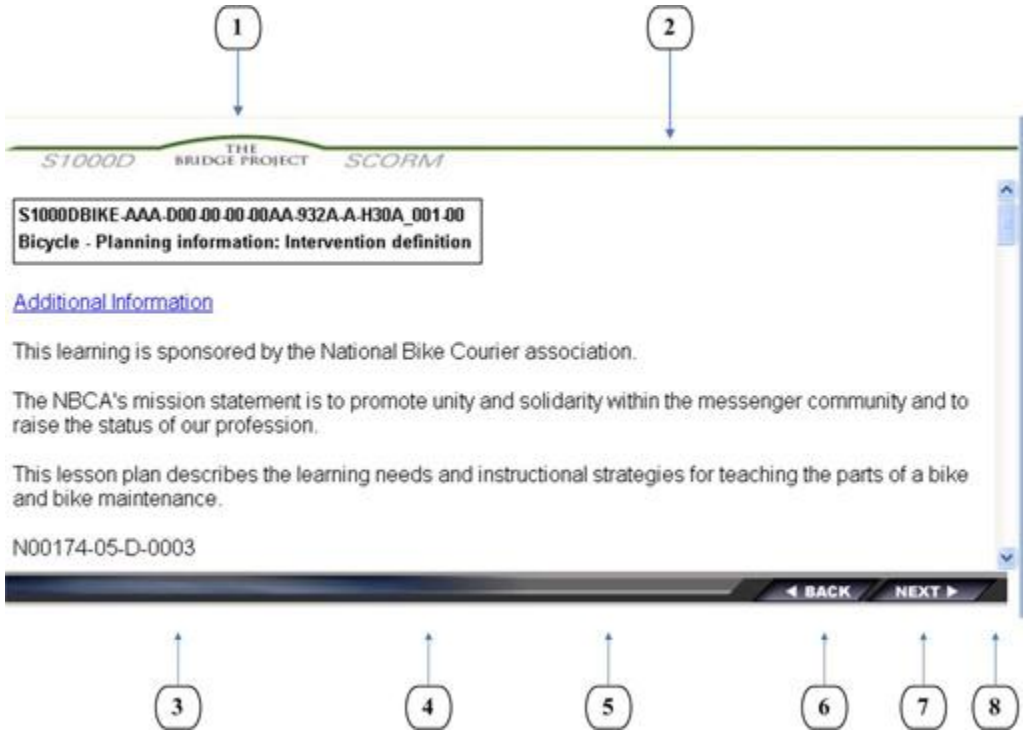

Fig. 6.1: Interface Elements

Key	Graphic Name	Graphic Description
1	toolkit_header_01.jpg	Bridge Project Header
2	toolkit_header_02.jpg	Header Background, stretches horizontally
3	toolkit_footer_01.jpg	Footer, left
4	toolkit_footer_02.jpg	Footer Background, stretches horizontally
5	toolkit_footer_03.jpg	Footer, middle

6	toolkit_footer_04.jpg	Back Button
7	toolkit_footer_05.jpg	Next Button
8	toolkit_footer_06.jpg	Footer Background

To replace these graphics with new ones, use this procedure:

1. Navigate to s1000DToolkit1.0/ViewerApplication/app/images and fetch the images you wish to replace.
2. Using these graphics as a template for content and size, create new graphics for your interface using your organization's logo, name, colors, etc.
3. Replace the existing graphics with the new ones in the app/images folder. The replacement graphics must have the exact same filename and type as the original.
4. Rebuild the JARs and republish the course using the procedure in section 6.5, Rebuilding JARs and Republishing Course.

6.1.1. Changing Interface Elements with Different Filename or Type

You may need to replace one type of graphic for another. For example, the original may be a .jpg and you may want to replace it with a .gif. If this is the case, you will need to modify html files located at s1000DToolkit1.0/ViewerApplication/app. The header images are referenced in the topz.htm file and the footer images are referenced in the navPage.htm file.

6.1.2. Modifying the Course Stylesheet

You may need to modify the stylesheet that governs the look and feel of the course. The name of this stylesheet is common.css which is found in "s1000DToolkit1.0\ViewerApplication\app."

6.1.3. Modifying Viewer Transformations

To modify the way Data Modules are rendered to XML, you will need to modify the viewer XSLT stylesheets. The viewer XSLT stylesheets are located in the "TrainingContent/app" directory. The root XSLT stylesheet, used to transform Data Modules to HTML at runtime, is the s1000d_4.xslt file. The subordinate stylesheets are linked in the root stylesheet through use of "xsl:include" statements. The subordinate stylesheets are referentially named to indicate their functionality within the rendering scheme.

6.2. Modifying the Mobile Web Application Output

The mobile web application utilizes the jQuery Mobile Framework. This framework provides many ways to easily customize the themes, colors, styles and icons used. More information can be found at http://jquerymobile.com/.

6.3. Modifying the PDF Output

The PDF module uses the Flying Saucer project. CSS files are used to transform the S1000D XML to PDF. The CSS file can be found in the "s1000DToolkit1.0\cssPDF" directory. Styling changes to the PDF output can be made to these CSS files. More information can be found at http://code.google.com/p/flying-saucer/.

6.4. Modifying the Apache Commons Configuration

You may need to modify the Controller class, which is the initial start of the Toolkit. Currently the toolkit has two different Controller classes: one for the command prompt (Controller.java) and one for the JavaGUI (ControllerJFrame.java).

You may need to manipulate the chain. The chain is configured in chain-config.xml which is found in "s1000DToolkit1.0\conf\bridge\toolkit."

See also: http://commons.apache.org/chain/cookbook.html

6.5. Rebuilding JARs and Republishing

After any modification is made, you will need to build new Toolkit jars and republish the output. For rebuilding and republishing, follow these steps:

1. Access the Command Prompt. It is easily accessed from the toolbar by clicking Start > Run. In the Open field type cmd and hit Enter.
2. Change the command prompt to the location of the toolkit. For example: c:\s1000DToolkit1.0
3. To build the command line jar, type “ant” and press Enter. To build the gui jar, type “ant guibuild” and press Enter.

Once the build is complete rerun the toolkit and publish the output. The changes will be evident in the published output.

7. Terms and Definitions

Term	Definition
Advanced Distributed Learning (ADL) Initiative	A DoD agency that leads the Federal participation with business and university groups, charged with developing consensus standards for training software and associated services. Source: http://www.adlnet.gov/About/Pages/adlinitiative.aspx.
Common Source Database (CSDB)	The CSDB is a collection of data modules. According to the S1000D specification, the CSDB is an information store and management tool for all objects required to produce the technical publications within projects. S1000D does not specify the design and implementation rules for a CSDB.
Data Module (DM)	The S1000D specification defines the Data Module in Chapter 3.2. Data Modules are small, reusable pieces of technical information. They are uniquely identified using a Data Module Code to allow ease of management and access in a database environment. Each data module
Extensible Stylesheet Language Transformation (XSLT)	Extensible Stylesheet Language Transformation (XSLT) is a declarative, XML-based language used for the transformation of XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document may be serialized (output) by the processor in standard XML syntax or in another format, such as HTML or plain text. XSLT is most often used to convert data between different XML schemas or to convert XML data into web pages or PDF documents. Source: http://en.wikipedia.org/wiki/XSLT.
HTML SCOs	HTML SCOs are the starting points for the SCOs created by the Toolkit for the SCORM Content Package. These are created during the transformation process in the SCO Builder module.
Identification and Status	The identification and status section gives all the identification elements required to address and control an S1000D data module. It also provides the status elements for information on the security, quality and technical status together with the applicability of the overall data module content. Source: S1000D Technical Specification, Chapter 3.9.5.1.
IMS Manifest (imsmanifest. xml)	The IMS Manifest is a required XML file at the root of every SCORM content package. It contains all organizational data, metadata and resource data for the SCORM course.
Learning Data Module (LDM)	S1000D supports technical training information development through the use of the learning data module. The Schema for this data module structures technical learning content and configures it to the system being instructed in the lessons. It also maintains the use of standard S1000D XML structures. By maintaining the common S1000D structures, reuse between technical data and its supporting learning content is possible without the need for conversion from other formats. Source: S1000D Technical Specification, Chapter 3.9.5.2.13.
Learning Management	A learning management system (commonly abbreviated as LMS) is a software application for the administration, documentation, tracking, and reporting of

System (LMS)	training programs, classroom and online events, e-learning programs, and training content. Source: http://en.wikipedia.org/wiki/Learning_management_system.
Learning Object Metadata (lom)	Learning Object Metadata is a data model, usually encoded in XML, used to describe a learning object and similar digital resources used to support learning. The purpose of learning object metadata is to support the reusability of learning objects, to aid discoverability, and to facilitate their interoperability, usually in the context of online learning management systems (LMS). Source: http://en.wikipedia.org/wiki/Learning_object_metadata.
Mobile Builder Module	The Toolkit runs the Mobile Builder Module to produce the mobile web application output.
Mobile Web Application	A mobile web application is content that can be hosted on any web server and accessed through a mobile device's web browser.
Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L))	The Office of the Undersecretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L)) is the title of a high-level civilian official in the United States Department of Defense. The Undersecretary of Defense for Acquisition, Technology and Logistics is the principal staff assistant and advisor to both the Secretary of Defense and the Deputy Secretary of Defense for all matters concerning the Acquisition, Technology and Logistics. This office has funded this Bridge project through its Reduction in Total Ownership Cost (RTOC) program. Source: http://en.wikipedia.org/wiki/Under_Secretary_of_Defense_for_Acquisition,_Technology_and_Logistics.
Organization	An organization is a level of training content hierarchy specified by the imsmanifest.xml.
PDF	Portable Document Format (PDF) is an open standard for document exchange.
PDF Builder Module	The Toolkit runs the PDF Builder Module to produce the PDF output. The current release does not include support for PDF output. PDF support will be delivered in a future release.
Post-Process Module	The Toolkit runs three modules to create the SCORM content package. The Post-Process Module is the last of the modules to run. The Post-Process Module copies the ADL Schemas, IEEE LOM Schemas and imsmanifest.xml (XML_SOURCE) to the SCORM content package (CP_PACKAGE) and finally zips the file up. Any cleanup of the Toolkit can happen here as well.
Pre-Process Module	The Toolkit runs three modules to create the SCORM Content Package. The Pre Process Module is the first of the modules to run. The preprocessing is where the mapping from SCPM to SCORM manifest file begins (XSLT transform).
Project Management Committee	The Project Management Committee is composed of the Source Forge project team members.
Reduction in Total	The Reduction in Total Ownership Costs (R-TOC) is a Department of Defense-(DoD-) wide effort to reduce total ownership costs that grew out of numerous

Ownership Cost (R-TOC)	reviews and discussions at Program Executive Officers'/Systems Command (PEO/SYSCOM) Commanders' conferences, the Defense Science Board, and others. Source: http://ve.ida.org/rtoc/rtoc.html.
Resource	A resource is a physical element of a SCORM Content Package that is specified by the imsmanifest.xml.
Resource Package	A Resource Package is a directory of data modules, and associated ICNs, that are referenced by a SCORM Content Package Module.
S1000D	S1000D is an international specification for the procurement and production of technical publications. It covers the planning and management, production, exchange, distribution and use of technical documentation that support the life cycle of any civil or military project. Projects include air, land and sea vehicles or equipments. It is an SGML/XML standard for preparing, managing, and using equipment maintenance and operations information. See http://www.S1000D.org.
S1000D Technical Data Specification	The S1000D Technical Data Specification is a document which provides a detailed, technical description of the S1000D standard. See http://www.S1000D.org.
S1000D Transformation Toolkit ("Toolkit")	The S1000D Transformation Toolkit is an open source tool that converts S1000D data into SCORM content packages.
SCO Builder Module	The Toolkit runs three modules to create the SCORM Content Package. The SCO Builder is the second of the modules to run. This is where the launchable learning resources are created. The output of the module is the set of html SCOs associated with each SCO. The CP_PACKAGE is created during this module.
scoEntry	The scoEntry is the XML element of the SCORM Content Package Module that identifies a module of training and maps to the SCO element of an imsmanifest.xml.
SCORM Content Package (CP)	The SCORM content package is the zip file that contains the training course created by the Toolkit. This output is SCORM conformant. This zip file is also known as a PIF file.
Sharable Content Object (SCO)	A Sharable Content Object (SCO) is the lowest level of interactive training content within a SCORM course. Since it is sharable, it can be re-used in any SCORM course. The IMS Manifest contains references to the SCOs of the course.
Sharable Content Package Module (SCPM)	The SCORM Content Package Module (SCPM) is a document in S1000D that identifies selected DMs for a SCORM course and maps DMs to the training objects of the course. Source: S1000D Technical Specification, Chapter 4.9.5.
Sharable Content	The Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based e-learning. It defines communications

Object Reference Model (SCORM)	between client side content and a host system called the run-time environment, which is commonly contained within a learning management system. SCORM also defines how content may be packaged into a transferable content package. This Toolkit is based on SCORM 2004 3rd Edition. See http://www.adlnet.gov/Technologies/scorm. All references to SCORM in this document means SCORM 2004 3rd Edition
Transformation	Transformation is the process of converting the aggregation structure of one file to another. The Toolkit uses transformation to convert the SCORM Content Package Module to the imsmanifest.xml.
Uniform Resource Name (URN) Map	A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that uses the urn scheme, and does not imply availability of the identified resource. Both URNs (names) and URLs (locators) are URIs, and a particular URI may be a name and a locator at the same time. Source: http://en.wikipedia.org/wiki/Uniform_Resource_Name.

8. Points of Contact

The Points of Contact for the S1000D-SCORM Bridge Toolkit Project are:

- Tyler Shumaker, Toolkit Task Manager, shumaket@ctc.com
- Steve Worsham, Developer, Worsham.Steve@idsi.com
- Schawn Thropp, Bridge Project Manager, ThroppS@ctc.com
- Sean Rushing, Document Writer, srushing@inmedius.com
- Mike Hamerly, Document Writer, mhamerly@inmedius.com
- Wayne Gafford, Bridge Project Government Manager, wayne.gafford@adlnet.gov