

Distance Learning - Deployment, Administration, Readiness and  
Training (DL DART)

Advanced Distributed Learning (ADL) Shareable Content Object  
Reference Model (SCORM) 2004 3<sup>rd</sup> Edition Conformance Test Suite  
v1.1.2

Report of Interim Progress

Provided for

The Army Distributed Learning Program

(TCM TADLP)



23 June 2014



## Introduction

This document summarizes recent changes to the Advanced Distributed Learning (ADL) Sharable Content Object Reference Model (SCORM) 2004 3<sup>rd</sup> Edition Conformance Test Suite (CTS) version 1.1.2 and provides a summary of potential courses of action (COAs) with regard to replacing the function of the *Resource Validator* application used in US Army SCORM testing.

## ADL SCORM 2004 3<sup>rd</sup> Edition Conformance Test Suite v1.1.2 Changes

The most recent change to the ADL SCORM 2004 3<sup>rd</sup> Edition Conformance Test Suite entailed a modification to the manner in which the CTS reads in the system environment variable used for application operations. The system environment variable is read in as an application variable for use within the test suite and identifies the location of the CTS on the system. The variable is used in application operations such as log generation and imported SCORM Content Package extraction.

Due to limitations in the InstallShield software available for generating an installation executable for the CTS, the system environment variable created during the installation process retains the backslash character appended to the file path. As a result, when the system environment variable is imported, an extra slash character is appended to the file paths used for operations such as launching SCOs and opening logs. This can cause a loss of functionality.

The CTS was modified to parse extraneous slash characters out if they are found at the end of the value sent to the application. This eliminates the potential for loss of functionality due to malformed file paths. See *Appendix A: getCTSEnvironmentVariable() Modifications* for further technical details.

## Appendix A: getCTSEnvironmentVariable() Modifications

The function in the CTS codebase responsible for reading the system environment variable into the application was modified to parse out slashes appended to the file path assigned to the variable. Shown in Figures 1 and 2 below is the function before and after the modification, with changes highlighted.

### **getCTSEnvironmentVariable() Function in CTSEnvironmentVariable.java (before)**

```
public static String getCTSEnvironmentVariable()
{
    if(dataAlreadyLookedUp)
        return mEnvironmentVariable;

    if (AuditorIndicator.ON)
    {
        mEnvironmentVariable = getValue(AT_ENV);
    }
    else
    {
        mEnvironmentVariable = getValue(ST_ENV);
    }

    //returns the environment variable
    return mEnvironmentVariable;
}
```

Figure 1: getCTSEnvironmentVariable() function before modification

### **getCTSEnvironmentVariable() Function in CTSEnvironmentVariable.java (after)**

```
public static String getCTSEnvironmentVariable()
{
    if(dataAlreadyLookedUp)
        return mEnvironmentVariable;

    if (AuditorIndicator.ON)
    {
        mEnvironmentVariable = getValue(AT_ENV);
    }
    else
    {
        mEnvironmentVariable = getValue(ST_ENV);
    }

    //returns the environment variable, removing trailing slash if present
    if ((mEnvironmentVariable.charAt(mEnvironmentVariable.length()-1)=='\\') ||
(mEnvironmentVariable.charAt(mEnvironmentVariable.length()-1)=='/'))
    {
        mEnvironmentVariable = mEnvironmentVariable.substring(0,
mEnvironmentVariable.length()-1);
    }
    else
    {
        //returns the environment variable
        return mEnvironmentVariable;
    }

    return mEnvironmentVariable;
}
```

Figure 2: getCTSEnvironmentVariable() function after modification