0:Thu Feb 10 20:00:59 CST 2011 - AICCComm - Trying to create MSXML2.XMLHTTP in VBScript  
1:Thu Feb 10 20:00:59 CST 2011 - AICCComm - intReCheckLoadedInterval=250  
2:Thu Feb 10 20:00:59 CST 2011 - AICCComm - intReCheckAttemptsBeforeTimeout=240  
3:Thu Feb 10 20:00:59 CST 2011 - AICCComm - IFrameLoaded   
4:Thu Feb 10 20:00:59 CST 2011 - AICCComm - In GetAICCURL  
5:Thu Feb 10 20:00:59 CST 2011 - GetQueryStringValue Element 'AICC\_URL' Not Found, Returning: empty string  
6:Thu Feb 10 20:00:59 CST 2011 - AICCComm - Querystring value =   
7:Thu Feb 10 20:00:59 CST 2011 - AICCComm - GetAICCURL returning:   
8:Thu Feb 10 20:00:59 CST 2011 - AICCComm - In DetectPreferredCommMethod, checking XMLHTTP  
9:Thu Feb 10 20:00:59 CST 2011 - AICCComm - Checking IFrame  
10:Thu Feb 10 20:00:59 CST 2011 - AICCComm - blnCanUseXMLHTTP=true  
11:Thu Feb 10 20:00:59 CST 2011 - AICCComm - blnCanUseIFrame=true  
12:Thu Feb 10 20:00:59 CST 2011 - ----------------------------------------  
13:Thu Feb 10 20:00:59 CST 2011 - ----------------------------------------  
14:Thu Feb 10 20:00:59 CST 2011 - In Start - Version: 3.3 Last Modified=02/10/2011 20:00:13  
15:Thu Feb 10 20:00:59 CST 2011 - Browser Info (Microsoft Internet Explorer 4.0 (compatible; MSIE 7.0; Windows NT 6.0; Trident/4.0; GTB0.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; InfoPath.2; .NET CLR 1.1.4322; .NET CLR 3.0.30729; .NET CLR 3.5.30729; .NET4.0C; OfficeLivePatch.1.3; OfficeLiveConnector.1.5; .NET4.0E; BRI/1; AskTbADAP/5.9.1.14019))  
16:Thu Feb 10 20:00:59 CST 2011 - URL: http://www.sippleoilcompany.com/hccscofspe/pluginfile.php/120/mod\_scorm/content/2/index\_lms.html  
17:Thu Feb 10 20:00:59 CST 2011 - ----------------------------------------  
18:Thu Feb 10 20:00:59 CST 2011 - ----------------------------------------  
19:Thu Feb 10 20:00:59 CST 2011 - In ClearErrorInfo  
20:Thu Feb 10 20:00:59 CST 2011 - GetQueryStringValue Element 'StandAlone' Not Found, Returning: empty string  
21:Thu Feb 10 20:00:59 CST 2011 - GetQueryStringValue Element 'ShowDebug' Not Found, Returning: empty string  
22:Thu Feb 10 20:00:59 CST 2011 - strStandAlone= strShowInteractiveDebug=  
23:Thu Feb 10 20:00:59 CST 2011 - Standard From Configuration File - SCORM  
24:Thu Feb 10 20:00:59 CST 2011 - Using Standard From Configuration File - SCORM  
25:Thu Feb 10 20:00:59 CST 2011 - In LMSStandardAPI strStandard=SCORM  
26:Thu Feb 10 20:00:59 CST 2011 - Calling Standard Initialize  
27:Thu Feb 10 20:00:59 CST 2011 - In SCORM\_Initialize  
28:Thu Feb 10 20:00:59 CST 2011 - In SCORM\_ClearErrorInfo  
29:Thu Feb 10 20:00:59 CST 2011 - Grabbing API  
30:Thu Feb 10 20:00:59 CST 2011 - In SCORM\_GrabAPI  
31:Thu Feb 10 20:00:59 CST 2011 - Searching with improved ADL algorithm  
32:Thu Feb 10 20:00:59 CST 2011 - In SCORM\_GetAPI  
33:Thu Feb 10 20:00:59 CST 2011 - SCORM\_GrabAPI, returning  
34:Thu Feb 10 20:00:59 CST 2011 - Unable to acquire SCORM API:  
35:Thu Feb 10 20:00:59 CST 2011 - SCORM\_objAPI=object  
36:Thu Feb 10 20:00:59 CST 2011 - In InitializeExecuted, blnSuccess=false, strErrorMessage=Error - unable to acquire LMS API, content may not play properly and results may not be recorded. Please contact technical support.  
37:Thu Feb 10 20:00:59 CST 2011 - ERROR - LMS Initialize Failed  
38:Thu Feb 10 20:00:59 CST 2011 - In DisplayError, strMessage=Error - unable to acquire LMS API, content may not play properly and results may not be recorde. Please contact technical support.

New start of new bug

// // SCORM 1.2 API Implementation // function SCORMapi1\_2() { // Standard Data Type Definition CMIString256 = '^[\\u0000-\\uffff]{0,255}$'; CMIString4096 = '^[\\u0000-\\uffff]{0,4096}$'; CMITime = '^([0-2]{1}[0-9]{1}):([0-5]{1}[0-9]{1}):([0-5]{1}[0-9]{1})(\.[0-9]{1,2})?$'; CMITimespan = '^([0-9]{2,4}):([0-9]{2}):([0-9]{2})(\.[0-9]{1,2})?$'; CMIInteger = '^\\d+$'; CMISInteger = '^-?([0-9]+)$'; CMIDecimal = '^-?([0-9]{0,3})(\.[0-9]{1,2})?$'; CMIIdentifier = '^[\\u0021-\\u007E]{0,255}$'; CMIFeedback = CMIString256; // This must be redefined CMIIndex = '[.\_](\\d+).'; // Vocabulary Data Type Definition CMIStatus = '^passed$|^completed$|^failed$|^incomplete$|^browsed$'; CMIStatus2 = '^passed$|^completed$|^failed$|^incomplete$|^browsed$|^not attempted$'; CMIExit = '^time-out$|^suspend$|^logout$|^$'; CMIType = '^true-false$|^choice$|^fill-in$|^matching$|^performance$|^sequencing$|^likert$|^numeric$'; CMIResult = '^correct$|^wrong$|^unanticipated$|^neutral$|^([0-9]{0,3})?(\.[0-9]{1,2})?$'; NAVEvent = '^previous$|^continue$'; // Children lists cmi\_children = 'core,suspend\_data,launch\_data,comments,objectives,student\_data,student\_preference,interactions'; core\_children = 'student\_id,student\_name,lesson\_location,credit,lesson\_status,entry,score,total\_time,lesson\_mode,exit,session\_time'; score\_children = 'raw,min,max'; comments\_children = 'content,location,time'; objectives\_children = 'id,score,status'; correct\_responses\_children = 'pattern'; student\_data\_children = 'mastery\_score,max\_time\_allowed,time\_limit\_action'; student\_preference\_children = 'audio,language,speed,text'; interactions\_children = 'id,objectives,time,type,correct\_responses,weighting,student\_response,result,latency'; // Data ranges score\_range = '0#100'; audio\_range = '-1#100'; speed\_range = '-100#100'; weighting\_range = '-100#100'; text\_range = '-1#1'; // The SCORM 1.2 data model var datamodel = { 'cmi.\_children':{'defaultvalue':cmi\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.\_version':{'defaultvalue':'3.4', 'mod':'r', 'writeerror':'402'}, 'cmi.core.\_children':{'defaultvalue':core\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.core.student\_id':{'defaultvalue':'Admin', 'mod':'r', 'writeerror':'403'}, 'cmi.core.student\_name':{'defaultvalue':'Sipple, James ', 'mod':'r', 'writeerror':'403'}, 'cmi.core.lesson\_location':{'defaultvalue':'', 'format':CMIString256, 'mod':'rw', 'writeerror':'405'}, 'cmi.core.credit':{'defaultvalue':'no-credit', 'mod':'r', 'writeerror':'403'}, 'cmi.core.lesson\_status':{'defaultvalue':'', 'format':CMIStatus, 'mod':'rw', 'writeerror':'405'}, 'cmi.core.entry':{'defaultvalue':'ab-initio', 'mod':'r', 'writeerror':'403'}, 'cmi.core.score.\_children':{'defaultvalue':score\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.core.score.raw':{'defaultvalue':'', 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.core.score.max':{'defaultvalue':'', 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.core.score.min':{'defaultvalue':'', 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.core.total\_time':{'defaultvalue':'00:00:00', 'mod':'r', 'writeerror':'403'}, 'cmi.core.lesson\_mode':{'defaultvalue':'', 'mod':'r', 'writeerror':'403'}, 'cmi.core.exit':{'defaultvalue':'', 'format':CMIExit, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.core.session\_time':{'format':CMITimespan, 'mod':'w', 'defaultvalue':'00:00:00', 'readerror':'404', 'writeerror':'405'}, 'cmi.suspend\_data':{'defaultvalue':'', 'format':CMIString4096, 'mod':'rw', 'writeerror':'405'}, 'cmi.launch\_data':{'defaultvalue':'', 'mod':'r', 'writeerror':'403'}, 'cmi.comments':{'defaultvalue':'', 'format':CMIString4096, 'mod':'rw', 'writeerror':'405'}, // deprecated evaluation attributes 'cmi.evaluation.comments.\_count':{'defaultvalue':'0', 'mod':'r', 'writeerror':'402'}, 'cmi.evaluation.comments.\_children':{'defaultvalue':comments\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.evaluation.comments.n.content':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMIString256, 'mod':'rw', 'writeerror':'405'}, 'cmi.evaluation.comments.n.location':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMIString256, 'mod':'rw', 'writeerror':'405'}, 'cmi.evaluation.comments.n.time':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMITime, 'mod':'rw', 'writeerror':'405'}, 'cmi.comments\_from\_lms':{'mod':'r', 'writeerror':'403'}, 'cmi.objectives.\_children':{'defaultvalue':objectives\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.objectives.\_count':{'mod':'r', 'defaultvalue':'0', 'writeerror':'402'}, 'cmi.objectives.n.id':{'pattern':CMIIndex, 'format':CMIIdentifier, 'mod':'rw', 'writeerror':'405'}, 'cmi.objectives.n.score.\_children':{'pattern':CMIIndex, 'mod':'r', 'writeerror':'402'}, 'cmi.objectives.n.score.raw':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.objectives.n.score.min':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.objectives.n.score.max':{'defaultvalue':'', 'pattern':CMIIndex, 'format':CMIDecimal, 'range':score\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.objectives.n.status':{'pattern':CMIIndex, 'format':CMIStatus2, 'mod':'rw', 'writeerror':'405'}, 'cmi.student\_data.\_children':{'defaultvalue':student\_data\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.student\_data.mastery\_score':{'defaultvalue':'80', 'mod':'r', 'writeerror':'403'}, 'cmi.student\_data.max\_time\_allowed':{'defaultvalue':'', 'mod':'r', 'writeerror':'403'}, 'cmi.student\_data.time\_limit\_action':{'defaultvalue':'', 'mod':'r', 'writeerror':'403'}, 'cmi.student\_preference.\_children':{'defaultvalue':student\_preference\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.student\_preference.audio':{'defaultvalue':'0', 'format':CMISInteger, 'range':audio\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.student\_preference.language':{'defaultvalue':'', 'format':CMIString256, 'mod':'rw', 'writeerror':'405'}, 'cmi.student\_preference.speed':{'defaultvalue':'0', 'format':CMISInteger, 'range':speed\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.student\_preference.text':{'defaultvalue':'0', 'format':CMISInteger, 'range':text\_range, 'mod':'rw', 'writeerror':'405'}, 'cmi.interactions.\_children':{'defaultvalue':interactions\_children, 'mod':'r', 'writeerror':'402'}, 'cmi.interactions.\_count':{'mod':'r', 'defaultvalue':'0', 'writeerror':'402'}, 'cmi.interactions.n.id':{'pattern':CMIIndex, 'format':CMIIdentifier, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.objectives.\_count':{'pattern':CMIIndex, 'mod':'r', 'defaultvalue':'0', 'writeerror':'402'}, 'cmi.interactions.n.objectives.n.id':{'pattern':CMIIndex, 'format':CMIIdentifier, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.time':{'pattern':CMIIndex, 'format':CMITime, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.type':{'pattern':CMIIndex, 'format':CMIType, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.correct\_responses.\_count':{'pattern':CMIIndex, 'mod':'r', 'defaultvalue':'0', 'writeerror':'402'}, 'cmi.interactions.n.correct\_responses.n.pattern':{'pattern':CMIIndex, 'format':CMIFeedback, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.weighting':{'pattern':CMIIndex, 'format':CMIDecimal, 'range':weighting\_range, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.student\_response':{'pattern':CMIIndex, 'format':CMIFeedback, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.result':{'pattern':CMIIndex, 'format':CMIResult, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'cmi.interactions.n.latency':{'pattern':CMIIndex, 'format':CMITimespan, 'mod':'w', 'readerror':'404', 'writeerror':'405'}, 'nav.event':{'defaultvalue':'', 'format':NAVEvent, 'mod':'w', 'readerror':'404', 'writeerror':'405'} }; // // Datamodel inizialization // var cmi = new Object(); cmi.core = new Object(); cmi.core.score = new Object(); cmi.objectives = new Object(); cmi.student\_data = new Object(); cmi.student\_preference = new Object(); cmi.interactions = new Object(); // deprecated evaluation attributes cmi.evaluation = new Object(); cmi.evaluation.comments = new Object(); // Navigation Object var nav = new Object(); for (element in datamodel) { if (element.match(/\.n\./) == null) { if ((typeof eval('datamodel["'+element+'"].defaultvalue')) != 'undefined') { eval(element+' = datamodel["'+element+'"].defaultvalue;'); } else { eval(element+' = "";'); } } } if (cmi.core.lesson\_status == '') { cmi.core.lesson\_status = 'not attempted'; } // // API Methods definition // var Initialized = false; function LMSInitialize (param) { errorCode = "0"; if (param == "") { if (!Initialized) { Initialized = true; errorCode = "0"; return "true"; } else { errorCode = "101"; } } else { errorCode = "201"; } return "false"; } this.connectPrereqCallback = { success: function(o) { scorm\_tree\_node = YAHOO.widget.TreeView.getTree('scorm\_tree'); if (o.responseText !== undefined) { //alert('got a response: ' + o.responseText); if (scorm\_tree\_node && o.responseText) { var hnode = scorm\_tree\_node.getHighlightedNode(); var hidx = null; if (hnode) { hidx = hnode.index + scorm\_tree\_node.getNodeCount(); } // all gone var root\_node = scorm\_tree\_node.getRoot(); while (root\_node.children.length > 0) { scorm\_tree\_node.removeNode(root\_node.children[0]); } } // make sure the temporary tree element is not there var el\_old\_tree = document.getElementById('scormtree123'); if (el\_old\_tree) { el\_old\_tree.parentNode.removeChild(el\_old\_tree); } var el\_new\_tree = document.createElement('div'); var pagecontent = document.getElementById("page-content"); el\_new\_tree.setAttribute('id','scormtree123'); el\_new\_tree.innerHTML = o.responseText; // make sure it doesnt show el\_new\_tree.style.display = 'none'; pagecontent.appendChild(el\_new\_tree) // ignore the first level element as this is the title var startNode = el\_new\_tree.firstChild.firstChild; if (startNode.tagName == 'LI') { // go back to the beginning startNode = el\_new\_tree; } //var sXML = new XMLSerializer().serializeToString(startNode); scorm\_tree\_node.buildTreeFromMarkup(startNode); var el = document.getElementById('scormtree123'); el.parentNode.removeChild(el); scorm\_tree\_node.expandAll(); scorm\_tree\_node.render(); if (hidx != null) { hnode = scorm\_tree\_node.getNodeByIndex(hidx); if (hnode) { hnode.highlight(); scorm\_layout\_widget = YAHOO.widget.Layout.getLayoutById('scorm\_layout'); var left = scorm\_layout\_widget.getUnitByPosition('left'); if (left.expanded) { hnode.focus(); } } } } }, failure: function(o) { // do some sort of error handling var sURL = "http://www.sippleoilcompany.com/hccscofspe" + "/mod/scorm/prereqs.php?a=5&scoid=38&attempt=1&mode=¤torg=&sesskey=4aeLOQkqjx"; //TODO: Enable this error handing correctly - avoiding issues when closing player MDL-23470 //alert('Prerequisites update failed - must restart SCORM player'); //window.location.href = sURL; } }; function LMSFinish (param) { errorCode = "0"; if (param == "") { if (Initialized) { Initialized = false; result = StoreData(cmi,true); if (nav.event != '') { if (nav.event == 'continue') { setTimeout('scorm\_get\_next();',500); } else { setTimeout('scorm\_get\_prev();',500); } } else { if (0 == 1) { setTimeout('scorm\_get\_next();',500); } } result = ('true' == result) ? 'true' : 'false'; errorCode = (result == 'true')? '0' : '101'; // trigger TOC update var sURL = "http://www.sippleoilcompany.com/hccscofspe" + "/mod/scorm/prereqs.php?a=5&scoid=38&attempt=1&mode=¤torg=&sesskey=4aeLOQkqjx"; YAHOO.util.Connect.asyncRequest('GET', sURL, this.connectPrereqCallback, null); return result; } else { errorCode = "301"; } } else { errorCode = "201"; } return "false"; } function LMSGetValue (element) { errorCode = "0"; if (Initialized) { if (element !="") { expression = new RegExp(CMIIndex,'g'); elementmodel = String(element).replace(expression,'.n.'); if ((typeof eval('datamodel["'+elementmodel+'"]')) != "undefined") { if (eval('datamodel["'+elementmodel+'"].mod') != 'w') { element = String(element).replace(expression, "\_$1."); elementIndexes = element.split('.'); subelement = 'cmi'; i = 1; while ((i < elementIndexes.length) && (typeof eval(subelement) != "undefined")) { subelement += '.'+elementIndexes[i++]; } if (subelement == element) { errorCode = "0"; return eval(element); } else { errorCode = "0"; // Need to check if it is the right errorCode } } else { errorCode = eval('datamodel["'+elementmodel+'"].readerror'); } } else { childrenstr = '.\_children'; countstr = '.\_count'; if (elementmodel.substr(elementmodel.length-childrenstr.length,elementmodel.length) == childrenstr) { parentmodel = elementmodel.substr(0,elementmodel.length-childrenstr.length); if ((typeof eval('datamodel["'+parentmodel+'"]')) != "undefined") { errorCode = "202"; } else { errorCode = "201"; } } else if (elementmodel.substr(elementmodel.length-countstr.length,elementmodel.length) == countstr) { parentmodel = elementmodel.substr(0,elementmodel.length-countstr.length); if ((typeof eval('datamodel["'+parentmodel+'"]')) != "undefined") { errorCode = "203"; } else { errorCode = "201"; } } else { errorCode = "201"; } } } else { errorCode = "201"; } } else { errorCode = "301"; } return ""; } function LMSSetValue (element,value) { errorCode = "0"; if (Initialized) { if (element != "") { expression = new RegExp(CMIIndex,'g'); elementmodel = String(element).replace(expression,'.n.'); if ((typeof eval('datamodel["'+elementmodel+'"]')) != "undefined") { if (eval('datamodel["'+elementmodel+'"].mod') != 'r') { expression = new RegExp(eval('datamodel["'+elementmodel+'"].format')); value = value+''; matches = value.match(expression); if (matches != null) { //Create dynamic data model element if (element != elementmodel) { elementIndexes = element.split('.'); subelement = 'cmi'; for (i=1;i < elementIndexes.length-1;i++) { elementIndex = elementIndexes[i]; if (elementIndexes[i+1].match(/^\d+$/)) { if ((typeof eval(subelement+'.'+elementIndex)) == "undefined") { eval(subelement+'.'+elementIndex+' = new Object();'); eval(subelement+'.'+elementIndex+'.\_count = 0;'); } if (elementIndexes[i+1] == eval(subelement+'.'+elementIndex+'.\_count')) { eval(subelement+'.'+elementIndex+'.\_count++;'); } if (elementIndexes[i+1] > eval(subelement+'.'+elementIndex+'.\_count')) { errorCode = "201"; } subelement = subelement.concat('.'+elementIndex+'\_'+elementIndexes[i+1]); i++; } else { subelement = subelement.concat('.'+elementIndex); } if ((typeof eval(subelement)) == "undefined") { eval(subelement+' = new Object();'); if (subelement.substr(0,14) == 'cmi.objectives') { eval(subelement+'.score = new Object();'); eval(subelement+'.score.\_children = score\_children;'); eval(subelement+'.score.raw = "";'); eval(subelement+'.score.min = "";'); eval(subelement+'.score.max = "";'); } if (subelement.substr(0,16) == 'cmi.interactions') { eval(subelement+'.objectives = new Object();'); eval(subelement+'.objectives.\_count = 0;'); eval(subelement+'.correct\_responses = new Object();'); eval(subelement+'.correct\_responses.\_count = 0;'); } } } element = subelement.concat('.'+elementIndexes[elementIndexes.length-1]); } //Store data if (errorCode == "0") { if ((typeof eval('datamodel["'+elementmodel+'"].range')) != "undefined") { range = eval('datamodel["'+elementmodel+'"].range'); ranges = range.split('#'); value = value\*1.0; if ((value >= ranges[0]) && (value <= ranges[1])) { eval(element+'=value;'); errorCode = "0"; return "true"; } else { errorCode = eval('datamodel["'+elementmodel+'"].writeerror'); } } else { if (element == 'cmi.comments') { cmi.comments = cmi.comments + value; } else { eval(element+'=value;'); } errorCode = "0"; return "true"; } } } else { errorCode = eval('datamodel["'+elementmodel+'"].writeerror'); } } else { errorCode = eval('datamodel["'+elementmodel+'"].writeerror'); } } else { errorCode = "201" } } else { errorCode = "201"; } } else { errorCode = "301"; } return "false"; } function LMSCommit (param) { errorCode = "0"; if (param == "") { if (Initialized) { result = StoreData(cmi,false); result = ('true' == result) ? 'true' : 'false'; errorCode = (result ==' true')? '0' : '101'; return result; } else { errorCode = "301"; } } else { errorCode = "201"; } return "false"; } function LMSGetLastError () { return errorCode; } function LMSGetErrorString (param) { if (param != "") { var errorString = new Array(); errorString["0"] = "No error"; errorString["101"] = "General exception"; errorString["201"] = "Invalid argument error"; errorString["202"] = "Element cannot have children"; errorString["203"] = "Element not an array - cannot have count"; errorString["301"] = "Not initialized"; errorString["401"] = "Not implemented error"; errorString["402"] = "Invalid set value, element is a keyword"; errorString["403"] = "Element is read only"; errorString["404"] = "Element is write only"; errorString["405"] = "Incorrect data type"; return errorString[param]; } else { return ""; } } function LMSGetDiagnostic (param) { if (param == "") { param = errorCode; } return param; } function AddTime (first, second) { var sFirst = first.split(":"); var sSecond = second.split(":"); var cFirst = sFirst[2].split("."); var cSecond = sSecond[2].split("."); var change = 0; FirstCents = 0; //Cents if (cFirst.length > 1) { FirstCents = parseInt(cFirst[1],10); } SecondCents = 0; if (cSecond.length > 1) { SecondCents = parseInt(cSecond[1],10); } var cents = FirstCents + SecondCents; change = Math.floor(cents / 100); cents = cents - (change \* 100); if (Math.floor(cents) < 10) { cents = "0" + cents.toString(); } var secs = parseInt(cFirst[0],10)+parseInt(cSecond[0],10)+change; //Seconds change = Math.floor(secs / 60); secs = secs - (change \* 60); if (Math.floor(secs) < 10) { secs = "0" + secs.toString(); } mins = parseInt(sFirst[1],10)+parseInt(sSecond[1],10)+change; //Minutes change = Math.floor(mins / 60); mins = mins - (change \* 60); if (mins < 10) { mins = "0" + mins.toString(); } hours = parseInt(sFirst[0],10)+parseInt(sSecond[0],10)+change; //Hours if (hours < 10) { hours = "0" + hours.toString(); } if (cents != '0') { return hours + ":" + mins + ":" + secs + '.' + cents; } else { return hours + ":" + mins + ":" + secs; } } function TotalTime() { total\_time = AddTime(cmi.core.total\_time, cmi.core.session\_time); return '&'+underscore('cmi.core.total\_time')+'='+encodeURIComponent(total\_time); } function CollectData(data,parent) { var datastring = ''; for (property in data) { if (typeof data[property] == 'object') { datastring += CollectData(data[property],parent+'.'+property); } else { element = parent+'.'+property; expression = new RegExp(CMIIndex,'g'); elementmodel = String(element).replace(expression,'.n.'); if (elementmodel != "cmi.core.session\_time") { if ((typeof eval('datamodel["'+elementmodel+'"]')) != "undefined") { if (eval('datamodel["'+elementmodel+'"].mod') != 'r') { elementstring = '&'+underscore(element)+'='+encodeURIComponent(data[property]); if ((typeof eval('datamodel["'+elementmodel+'"].defaultvalue')) != "undefined") { if (eval('datamodel["'+elementmodel+'"].defaultvalue') != data[property] || eval('typeof(datamodel["'+elementmodel+'"].defaultvalue)') != typeof(data[property])) { datastring += elementstring; eval('datamodel["'+elementmodel+'"].defaultvalue=data[property];'); } } else { datastring += elementstring; eval('datamodel["'+elementmodel+'"].defaultvalue=data[property];'); } } } } } } return datastring; } function StoreData(data,storetotaltime) { if (storetotaltime) { if (cmi.core.lesson\_status == 'not attempted') { cmi.core.lesson\_status = 'completed'; } if (cmi.core.lesson\_mode == 'normal') { if (cmi.core.credit == 'credit') { if (cmi.student\_data.mastery\_score != '' && cmi.core.score.raw != '') { if (parseFloat(cmi.core.score.raw) >= parseFloat(cmi.student\_data.mastery\_score)) { cmi.core.lesson\_status = 'passed'; } else { cmi.core.lesson\_status = 'failed'; } } } } if (cmi.core.lesson\_mode == 'browse') { if (datamodel['cmi.core.lesson\_status'].defaultvalue == '' && cmi.core.lesson\_status == 'not attempted') { cmi.core.lesson\_status = 'browsed'; } } datastring = CollectData(data,'cmi'); datastring += TotalTime(); } else { datastring = CollectData(data,'cmi'); } datastring += '&attempt=1'; datastring += '&scoid=38'; var myRequest = NewHttpReq(); //alert('going to:' + "http://www.sippleoilcompany.com/hccscofspe/mod/scorm/datamodel.php" + "id=&a=5&sesskey=4aeLOQkqjx"+datastring); result = DoRequest(myRequest,"http://www.sippleoilcompany.com/hccscofspe/mod/scorm/datamodel.php","id=&a=5&sesskey=4aeLOQkqjx"+datastring); results = String(result).split('\n'); errorCode = results[1]; return results[0]; } this.LMSInitialize = LMSInitialize; this.LMSFinish = LMSFinish; this.LMSGetValue = LMSGetValue; this.LMSSetValue = LMSSetValue; this.LMSCommit = LMSCommit; this.LMSGetLastError = LMSGetLastError; this.LMSGetErrorString = LMSGetErrorString; this.LMSGetDiagnostic = LMSGetDiagnostic; } var API = new SCORMapi1\_2(); var errorCode = "0"; function underscore(str) { str = String(str).replace(/.N/g,"."); return str.replace(/\./g,"\_\_"); }

Start new

0:Thu Feb 10 20:14:55 CST 2011 - AICCComm - Trying to create MSXML2.XMLHTTP in VBScript  
1:Thu Feb 10 20:14:55 CST 2011 - AICCComm - intReCheckLoadedInterval=250  
2:Thu Feb 10 20:14:55 CST 2011 - AICCComm - intReCheckAttemptsBeforeTimeout=240  
3:Thu Feb 10 20:14:55 CST 2011 - AICCComm - IFrameLoaded   
4:Thu Feb 10 20:14:55 CST 2011 - AICCComm - In GetAICCURL  
5:Thu Feb 10 20:14:55 CST 2011 - GetQueryStringValue Element 'AICC\_URL' Not Found, Returning: empty string  
6:Thu Feb 10 20:14:55 CST 2011 - AICCComm - Querystring value =   
7:Thu Feb 10 20:14:55 CST 2011 - AICCComm - GetAICCURL returning:   
8:Thu Feb 10 20:14:55 CST 2011 - AICCComm - In DetectPreferredCommMethod, checking XMLHTTP  
9:Thu Feb 10 20:14:55 CST 2011 - AICCComm - Checking IFrame  
10:Thu Feb 10 20:14:55 CST 2011 - AICCComm - blnCanUseXMLHTTP=true  
11:Thu Feb 10 20:14:55 CST 2011 - AICCComm - blnCanUseIFrame=true  
12:Thu Feb 10 20:14:55 CST 2011 - ----------------------------------------  
13:Thu Feb 10 20:14:55 CST 2011 - ----------------------------------------  
14:Thu Feb 10 20:14:55 CST 2011 - In Start - Version: 3.3 Last Modified=02/10/2011 20:10:07  
15:Thu Feb 10 20:14:55 CST 2011 - Browser Info (Microsoft Internet Explorer 4.0 (compatible; MSIE 7.0; Windows NT 6.0; Trident/4.0; GTB0.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; InfoPath.2; .NET CLR 1.1.4322; .NET CLR 3.0.30729; .NET CLR 3.5.30729; .NET4.0C; OfficeLivePatch.1.3; OfficeLiveConnector.1.5; .NET4.0E; BRI/1; AskTbADAP/5.9.1.14019))  
16:Thu Feb 10 20:14:55 CST 2011 - URL: http://www.sippleoilcompany.com/hccscofspe/pluginfile.php/129/mod\_scorm/content/1/index\_lms.html  
17:Thu Feb 10 20:14:55 CST 2011 - ----------------------------------------  
18:Thu Feb 10 20:14:55 CST 2011 - ----------------------------------------  
19:Thu Feb 10 20:14:55 CST 2011 - In ClearErrorInfo  
20:Thu Feb 10 20:14:55 CST 2011 - GetQueryStringValue Element 'StandAlone' Not Found, Returning: empty string  
21:Thu Feb 10 20:14:55 CST 2011 - GetQueryStringValue Element 'ShowDebug' Not Found, Returning: empty string  
22:Thu Feb 10 20:14:55 CST 2011 - strStandAlone= strShowInteractiveDebug=  
23:Thu Feb 10 20:14:55 CST 2011 - Standard From Configuration File - SCORM  
24:Thu Feb 10 20:14:55 CST 2011 - Using Standard From Configuration File - SCORM  
25:Thu Feb 10 20:14:55 CST 2011 - In LMSStandardAPI strStandard=SCORM  
26:Thu Feb 10 20:14:55 CST 2011 - Calling Standard Initialize  
27:Thu Feb 10 20:14:55 CST 2011 - In SCORM\_Initialize  
28:Thu Feb 10 20:14:55 CST 2011 - In SCORM\_ClearErrorInfo  
29:Thu Feb 10 20:14:55 CST 2011 - Grabbing API  
30:Thu Feb 10 20:14:55 CST 2011 - In SCORM\_GrabAPI  
31:Thu Feb 10 20:14:55 CST 2011 - Searching with improved ADL algorithm  
32:Thu Feb 10 20:14:55 CST 2011 - In SCORM\_GetAPI  
33:Thu Feb 10 20:14:55 CST 2011 - SCORM\_GrabAPI, returning  
34:Thu Feb 10 20:14:55 CST 2011 - Unable to acquire SCORM API:  
35:Thu Feb 10 20:14:55 CST 2011 - SCORM\_objAPI=object  
36:Thu Feb 10 20:14:55 CST 2011 - In InitializeExecuted, blnSuccess=false, strErrorMessage=Error - unable to acquire LMS API, content may not play properly and results may not be recorded. Please contact technical support.  
37:Thu Feb 10 20:14:55 CST 2011 - ERROR - LMS Initialize Failed  
38:Thu Feb 10 20:14:55 CST 2011 - In DisplayError, strMessage=Error - unable to acquire LMS API, content may not play properly and results may not be recorded. Please contact technical support.