

Submitting Test Results to CGI/ASP Script

Introduction

Lectora has the ability to capture student information such as names, email addresses, test answers, and test results so that you can manage your courses and your students more efficiently and effectively. Connecting Lectora to your database requires CGI or ASP scripting, which directs the data to a database. Because databases are different, custom CGI/ASP scripting must be written for each Lectora-to-Database connection.

The information on the following pages is intended to help you design a CGI/ASP script. Combining the information offered here with the expertise of your database manager and IT or ISP manager, you should be able to develop an effective connection between Lectora and your database.

Please remember this is an example of CGI and ASP script. A programmer knowledgeable in CGI/ASP Scripting and Database integration over the Internet/Intranet should perform the integration between Lectora and the database.

1. Variables in Lectora

This page shows Lectora's variable structure

2. Add a CGI/ASP Script to Lectora

This page describes how to integrate a CGI Script in to Lectora for both tests and forms.

3. Script Examples

This page gives an example of a CGI and ASP Script which receives test results.

4. Connecting to a Database

This section explains how to use scripts with a database and includes an MS Access and ASP example.

Please keep in mind that the sample script offered is just an example. It is a model on which you can base your own custom-written CGI/ASP script.

If you have any questions about this information contained here, please feel welcome to call Trivantis Technical Services at (513)852-6700.

1. Variables in Lectora

Lectora provides the ability to submit test and form results to a CGI/ASP application. This is done by selecting the Submit Test to CGI/ASP Program option in a test's properties dialog and also in the Submit form to CGI/ASP program in the form properties, and subsequently specifying a CGI/ASP program to which results will be sent in the area provided. Use Name Value Pairs to get the values from Lectora.

Parameters which are sent to a CGI/ASP application via **test submission**:

<u>Name</u>	<u>Value</u>
Name	The students name
Email	The students email address
TestName	The name of the test
NumQuestions	The number of questions in the test
PassingGrade	The value specified in the Lowest Passing score field on the Results tab of the Test's properties. If the output of the test is sent to XML, then the passing score itself is not sent, but the result of passing or failing is represented by the XML notation <passed /> or <failed />.
Score	The total score of the test (if it could be scored)
TrueFalse	The number of True/False questions correct/in the test
MultipleChoice	The number of multiple choice questions correct/in the test
XML	An XML formatted collection of the above parameters (see XML example below)

The next section would iterate for as many questions as were in the test. In each of the following the X is replaced by the actual number of the question (i.e. Question1, Answer1):

<u>Name</u>	<u>Value</u>
QuestionX	The text of the question
AnswerX	The students answer to the question
CorrectAnswerX	The correct answer to the question
QuestionTypeX	The type of the question (TRUEFALSE=1, MULTCHOICE=2, SHORTANSWER=3, ESSAY=4)

Additional Variables may be sent by selecting, "include all variable values upon submission":

<u>Name</u>	<u>Value</u>
BrowserType	The name of the browser used to view the title. When in run/preview mode, this value will be <i>Lectora</i> , when in a CD publish this value will be <i>LectoraViewer</i> .
CurrentChapterName, CurrentPageName, CurrentSectionName, CurrentTitleName	Displays the current name of the title, chapter, section, or page the user is viewing (respectfully.) This value is set to <i>unknown</i> if not currently being used.
CurrentDate	Contains today's date in the form Month, Day, Year. For example: January 1, 2003.
CurrentTime	Has the current time in the form hh:mm AM/PM. For example: 2:21 PM.

ElapsedTime	The total time the user has been viewing the title.
PageInChapter, PageInTitle	The current page number with the chapter/title the user is on
PagesInChapter, PagesInTitle	Total number of pages in the current title/chapter
Platform	The operating system the user is running the title from. (Win32, MacPPC, Windows NT)
PublishDate	The date that the title was published.

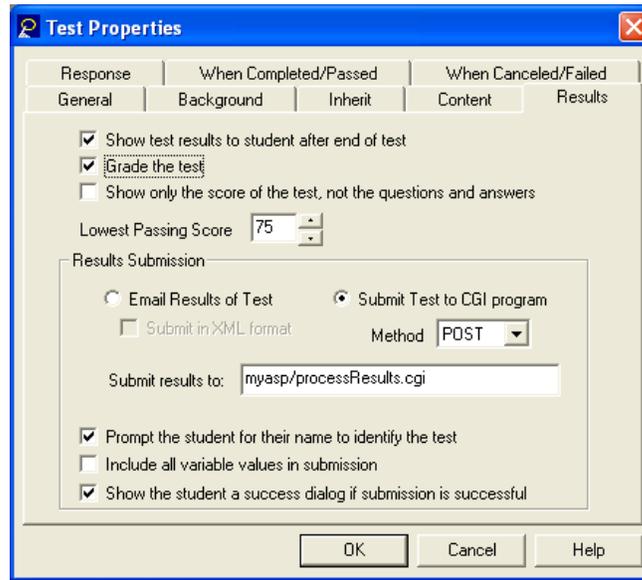
Note that any variables you create will also be sent as well, if including all variables on submission.

2. Add CGI/ASP to Lectora

Many problems stem from people not knowing what to put in the "submit results to" field on the test properties. You may send the results to a specific web url using "http://" and then type the full path to script program.

If you include your script with the rest of your title, you may simply use a relative path. For example, if you were using ASP, and the test processing page was called "processResults.asp", and it was in a directory called "myasp", the submit results to line would say "myasp/processResults.asp".

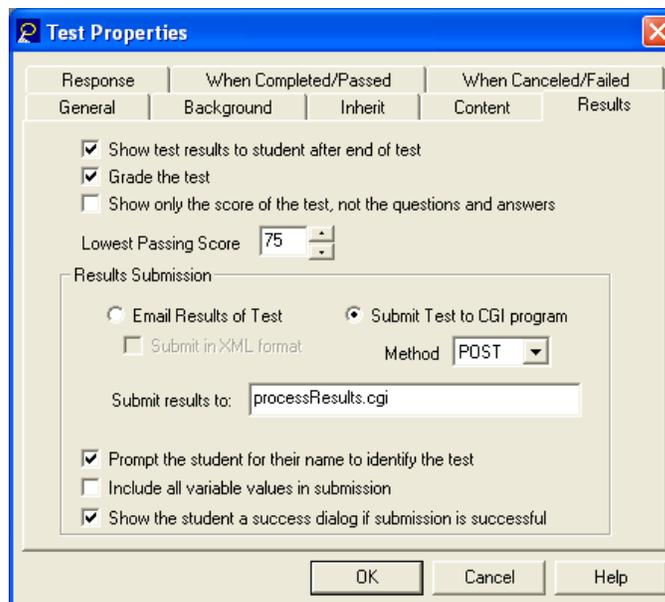
Through a test



The screenshot shows the "Test Properties" dialog box with the "Results" tab selected. The "Results Submission" section is expanded, showing the following settings:

- Show test results to student after end of test
- Grade the test
- Show only the score of the test, not the questions and answers
- Lowest Passing Score: 75
- Results Submission:
 - Email Results of Test
 - Submit Test to CGI program
 - Submit in XML format
 - Method: POST
 - Submit results to: myasp/processResults.cgi
- Prompt the student for their name to identify the test
- Include all variable values in submission
- Show the student a success dialog if submission is successful

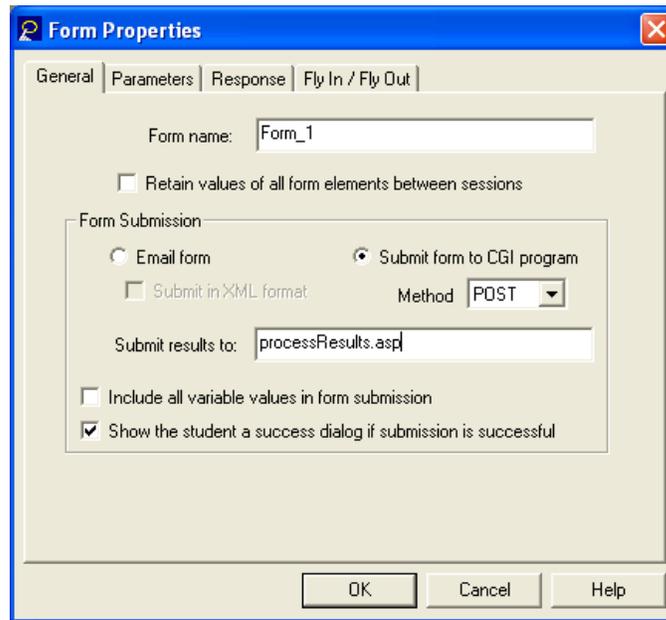
Or if you were using ASP, and their test processing page was called "processResults.asp", and the page was at the same level as the index.html page, the submit results to line would say.



The screenshot shows the "Test Properties" dialog box with the "Results" tab selected. The "Results Submission" section is expanded, showing the following settings:

- Show test results to student after end of test
- Grade the test
- Show only the score of the test, not the questions and answers
- Lowest Passing Score: 75
- Results Submission:
 - Email Results of Test
 - Submit Test to CGI program
 - Submit in XML format
 - Method: POST
 - Submit results to: processResults.cgi
- Prompt the student for their name to identify the test
- Include all variable values in submission
- Show the student a success dialog if submission is successful

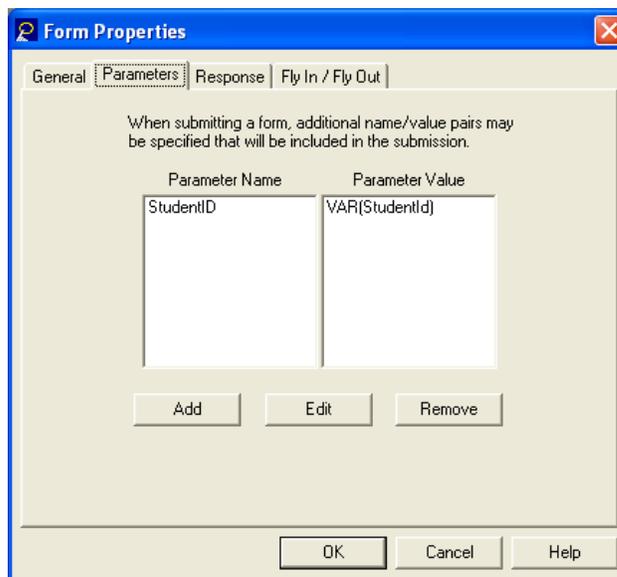
Through a form



The screenshot shows the 'Form Properties' dialog box with the 'General' tab selected. The 'Form name' is 'Form_1'. The 'Retain values of all form elements between sessions' checkbox is unchecked. The 'Form Submission' section has 'Submit form to CGI program' selected, with 'Submit in XML format' unchecked and 'Method' set to 'POST'. The 'Submit results to' field contains 'processResults.asp'. The 'Include all variable values in form submission' checkbox is unchecked, and the 'Show the student a success dialog if submission is successful' checkbox is checked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

The above example will submit all variables associated with the form (radio buttons, entry fields, ect) to processResults.asp, which is in the same folder as the rest of the title files.

If you wish to include other values with the form you may click on the “Parameters” tab and add each value individually. See the example below.



The screenshot shows the 'Form Properties' dialog box with the 'Parameters' tab selected. The text reads: 'When submitting a form, additional name/value pairs may be specified that will be included in the submission.' Below this is a table with two columns: 'Parameter Name' and 'Parameter Value'. The table contains one entry: 'StudentID' in the Parameter Name column and 'VAR(StudentID)' in the Parameter Value column. Below the table are 'Add', 'Edit', and 'Remove' buttons. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

Parameter Name	Parameter Value
StudentID	VAR(StudentID)

Here, we will send the parameter StudentID and it will contain the value of the variable StudentID from the title. This is done by setting it's value to VAR(StudentID). You may set any parameter to a Lectora variable using the VAR(*LectoraVariableName*) function.

3. Script Example

Example ASP Script

**** -- This script is for example purposes only -- ****

```
<%@ Language=VBScript %>
<%
'Get the parameters posted from the test'
testname=Request.form("TestName")
score=Request.form("Score")
user=Request.form("name")
email=Request.form("email")
numQuestions=Request.form("NumQuestions")
passingGrade=Request.form("PassingGrade")
trueFalse=Request.form( "TrueFalse" )
multipleChoice=Request.form( "MultipleChoice" )

'Validate that this is actually from a Lectora test'
if testname="" Or score="" Or user="" Or email="" Or numQuestions="" Or passingGrade="" Or
trueFalse="" Or multipleChoice="" then
    Response.Write "<html>"
    Response.Write "<head><title>Failure</title></head>"
    Response.Write "<body>"
    Response.Write "STATUS=500"
    Response.Write "<br>"
    Response.Write "Could not parse test results due to a parameter error."
    Response.Write "</body></html>"
else
    'Write the results to a file'
    'This could be a database or any kind of object store, but'
    'to keep it simple, we will just use a flat text file'
    fileName = "C:\" & testname & ".log"

    'Open the results file for append'
    Const ForReading = 1, ForWriting = 2, ForAppending = 8
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    Set objInFile = objFSO.OpenTextFile( fileName, ForAppending, True )

    'Write the results'
    objInFile.WriteLine( Date & ", " & Time & ", " & user & ", " & email & ", " & score )

    'Close results file'
    objInFile.Close
    Set objInFile = Nothing
    Set objFSO = Nothing

    'The response is optional, it is good for debugging'
    Response.Write "<html>"
    Response.Write "<head><title>Success</title></head>"
    Response.Write "<body>"
    Response.Write "STATUS=200"
    Response.Write "<br>"
    Response.Write "Success."
    Response.Write "</body></html>"
end if
%>
```

Example CGI Script

```
#!/perl
use CGI;
$q = new CGI;
#get the parameters passed to the script
#####
$name = $q->param('name');
$testName = $q->param('TestName');
$numQuestions = $q->param('NumQuestions');
$passingGrade = $q->param('PassingGrade');
$score = $q->param('Score');
if( $testName eq "" || $numQuestions eq "" )
{
    print "Content-type: text/html\n\n";
    print "<html>";
    print "<head><title>Failure</title></head>";
    print "<body>";
    print "STATUS=500";
    print "<br>";
    print "Could not parse test results due to a parameter error.";
    print "</body></html>";
    exit 0;
}
#####
#get the current time
#####
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
$year = $year + 1900;
$mon = $mon + 1;
$currtime = "$year-$mon-$mday $hour:$min:$sec";

#####
#opens test data file for append and write the data
#####
$testName .= ".log";
open( TESTDATA,">>c:\\$testName" ) or die;
print TESTDATA "$currtime $name $numQuestions $passingGrade $score\n";

#####
# Older courses produced by Lectora used a zero based index for the questions (i.e. Question0 is the first
question)'
# Newer courses are one based (i.e. Question1 is the first question)'
# determine which one it is'
#####
$startindex = 0;
$temp = $q->param('Question0');
if( $temp eq "" )
{
    $startindex = 1;
}
$index = $startindex;

#####
#Write out all of the questions and answers
#####
while( $index < ($startindex + $numQuestions) )
{
    $qst = "Question$index";
    $ans = "Answer$index";
```

```
$qval = $q->param( $qst );
$aval = $q->param( $ans );
print TESTDATA "$qst: $qval\n";
print TESTDATA "$ans: $aval\n";
$index += 1;
}
```

```
close TESTDATA;
#####
#reply that it worked
#####
print $q->header();
print "STATUS=200\r\n";
```

4. Connecting to a Database

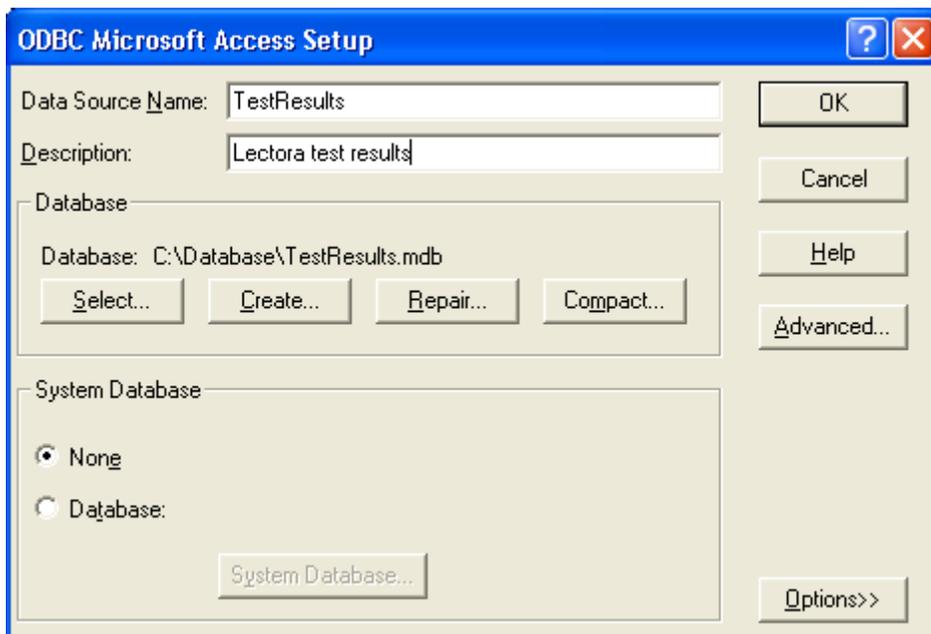
Now that the basics of script files have been covered, a more advanced type script may be used to store the results inside of a database. The included zip file **'Database_example.zip'** contains a sample Access database and ASP scripts to store test results.

There are 3 files here. There is an asp script called **saveresults.asp** which takes the test results and stores them to the database. This is the file you link to inside of your test properties (as mentioned in the 'through a test' section above.)

The second asp script called **viewresults.asp** can be viewed in the web browser and will show all the data from the database inside a table. Both of these scripts need to be placed on the web server.

The next step is to take the database file **TestResults.mdb** and place it in a new directory on the web server called c:\database. Now you must link to the database using ODBC.

- Go into the control panel, click administrative tools, click Data Sources (ODBC)
- Go into the "system DSN" tab and click add new data source
- Select "driver do Microsoft Access .mdb" as the driver
- Click finish, for data source name use "TestResults"
- Select the database file under c:\database\ TestResults.mdb
- Click OK



You should now have a simple working database to store test results.