# Lertap 5 Macros for Pearson VUE Data

Larry Nelson

--- Lertap.com ---

Document date: 24 November 2014

---

*"VUE" was a company formed in 1994 as "Virtual University Examinations".*

*In 1997 NCS, National Computer Systems, purchased VUE. (NCS bought the rights to use the computer code seen in the first personal-computer versions of Lertap from us in 1984[1]. NCS' MicroTest1 and MicroSurvey1 products were at least partially inspired by Lertap, and came to be freely distributed with NCS Scantron optical scanners.)*

*In 2000, Pearson PLC, a London-based company, acquired NCS, and Pearson VUE was born.*

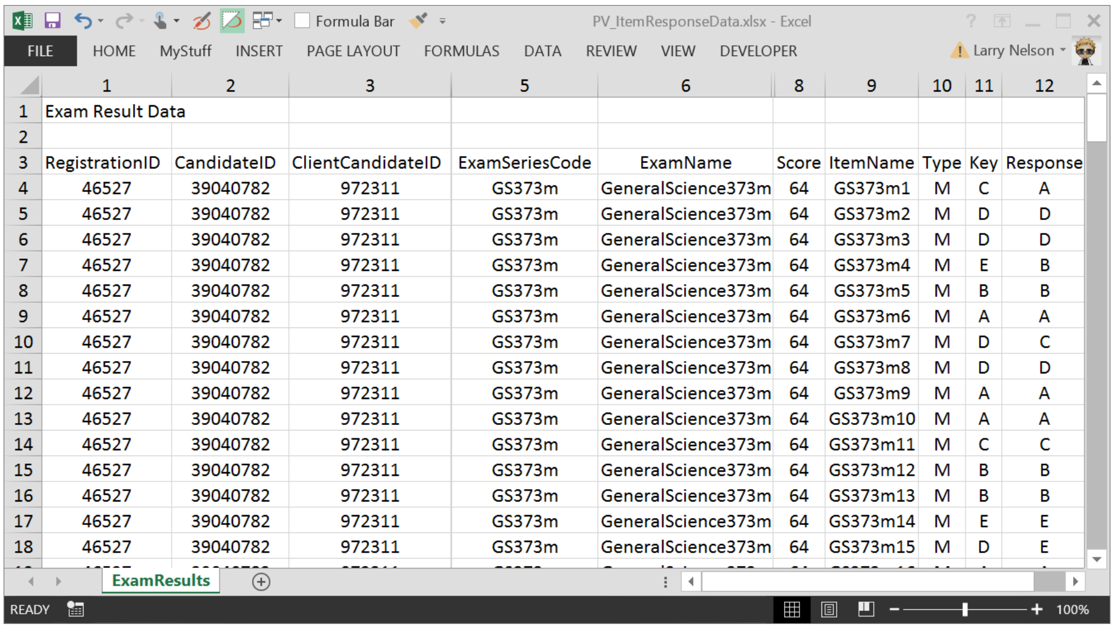
*Pearson VUE, or "PV", specialises in offering online testing services. Some Lertap users have contracted with PV to deliver their exams; these are administered by PV, often on a global basis, with results returned to users as specially-formatted computer files. In turn, users have then requested assistance in developing small computer programs, or "macros", which will reformat PV-collected results so that they may be analysed using Lertap 5.*

*To date we have created an extensive suite of macros for users of PV's Client Data Interchange (CDI) services. The macros unpack compressed PV data files from a variety of online certification tests and assemble a battery of Lertap-ready Excel workbooks.*

*We have also worked on smaller macros designed to reformat PV-created Excel workbooks of results so that the item response data they contain are ready for Lertap 5. The macro described below is an example based on a modified VUE CDI "Item Result File". It's a free macro, not copyrighted. Readers may find that the macro may suit their own needs, or may be adapted locally, at their own site, without too much ado.*

[1] At that time we were trading as "ASS", Antipodes Software Systems, a business first registered in New Zealand.

## An Excel workbook from Pearson VUE CDI

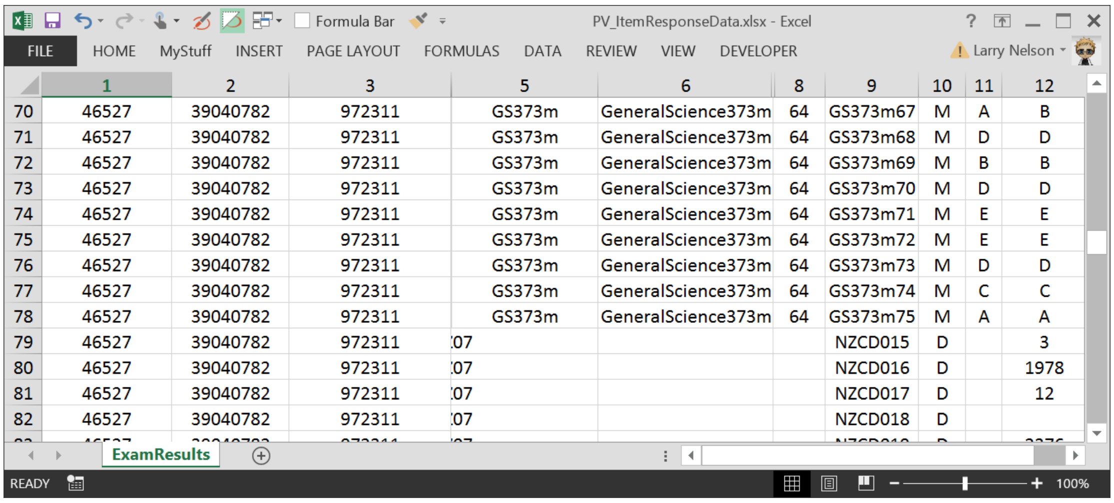


|  | 1 | 2 | 3 | 5 | 6 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Exam Result Data |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |
| 3 | RegistrationID | CandidateID | ClientCandidateID | ExamSeriesCode | ExamName | Score | ItemName | Type | Key | Response |
| 4 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m1 | M | C | A |
| 5 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m2 | M | D | D |
| 6 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m3 | M | D | D |
| 7 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m4 | M | E | B |
| 8 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m5 | M | B | B |
| 9 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m6 | M | A | A |
| 10 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m7 | M | D | C |
| 11 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m8 | M | D | D |
| 12 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m9 | M | A | A |
| 13 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m10 | M | A | A |
| 14 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m11 | M | C | C |
| 15 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m12 | M | B | B |
| 16 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m13 | M | B | B |
| 17 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m14 | M | E | E |
| 18 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m15 | M | D | E |

The screen shot above displays selected columns from a sample VUE worksheet.

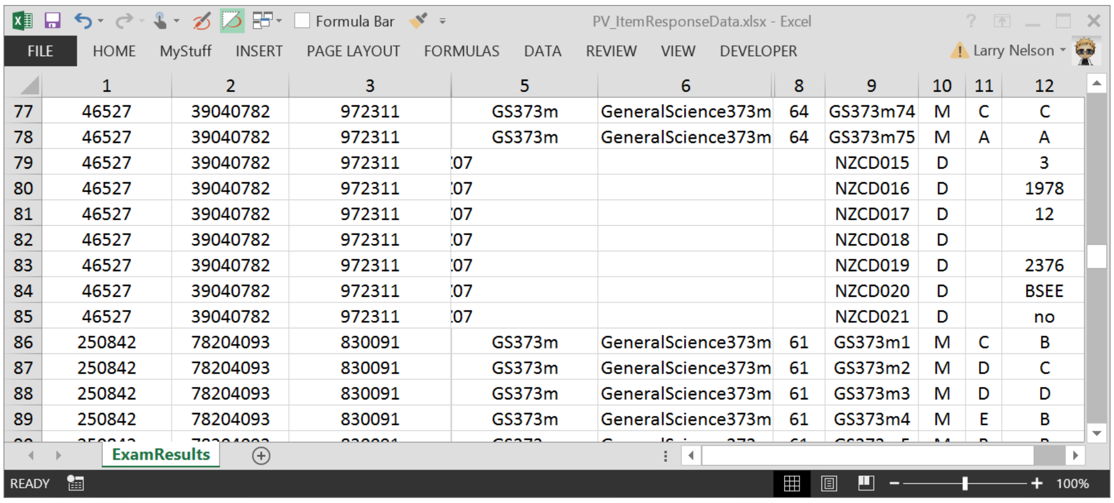
In this case, a 75-item test was taken online by a few hundred students, or "candidates". The first three rows of the sheet are reserved for header information, with the second row having no content at all.

We can see a lot of repeated information in these rows, something I have found to characterise VUE datasets of all sorts, not just Excel worksheets. In this example, the information in most of the columns is constant for any given candidate – only columns 9, 11, and 12 change as we look down them.



|  | 1 | 2 | 3 | 5 | 6 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|
| 70 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m67 | M | A | B |
| 71 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m68 | M | D | D |
| 72 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m69 | M | B | B |
| 73 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m70 | M | D | D |
| 74 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m71 | M | E | E |
| 75 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m72 | M | E | E |
| 76 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m73 | M | D | D |
| 77 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m74 | M | C | C |
| 78 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m75 | M | A | A |
| 79 | 46527 | 39040782 | 972311 | :07 |  |  | NZCD015 | D |  | 3 |
| 80 | 46527 | 39040782 | 972311 | :07 |  |  | NZCD016 | D |  | 1978 |
| 81 | 46527 | 39040782 | 972311 | :07 |  |  | NZCD017 | D |  | 12 |
| 82 | 46527 | 39040782 | 972311 | :07 |  |  | NZCD018 | D |  |  |

The shot above displays rows 70 through 82 of the worksheet. Data for ClientCandidateID 972311 are still on show, but note the change at row 79: the "Key" field in column 11 has ended – this signals the extent of the multiple-choice test items, those to be scored by Lertap. Also note, if you haven't already, that the candidate's response to each item is recorded in column 12, with "ItemNames" in column 9.



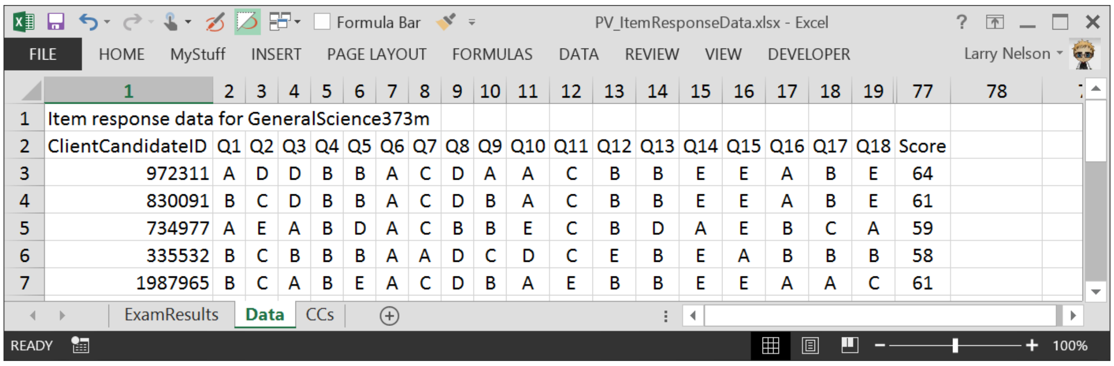
| | 1 | 2 | 3 | 5 | 6 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|
| 77 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m74 | M | C | C |
| 78 | 46527 | 39040782 | 972311 | GS373m | GeneralScience373m | 64 | GS373m75 | M | A | A |
| 79 | 46527 | 39040782 | 972311 | :07 | | | NZCD015 | D | | 3 |
| 80 | 46527 | 39040782 | 972311 | :07 | | | NZCD016 | D | | 1978 |
| 81 | 46527 | 39040782 | 972311 | :07 | | | NZCD017 | D | | 12 |
| 82 | 46527 | 39040782 | 972311 | :07 | | | NZCD018 | D | | |
| 83 | 46527 | 39040782 | 972311 | :07 | | | NZCD019 | D | | 2376 |
| 84 | 46527 | 39040782 | 972311 | :07 | | | NZCD020 | D | | BSEE |
| 85 | 46527 | 39040782 | 972311 | :07 | | | NZCD021 | D | | no |
| 86 | 250842 | 78204093 | 830091 | GS373m | GeneralScience373m | 61 | GS373m1 | M | C | B |
| 87 | 250842 | 78204093 | 830091 | GS373m | GeneralScience373m | 61 | GS373m2 | M | D | C |
| 88 | 250842 | 78204093 | 830091 | GS373m | GeneralScience373m | 61 | GS373m3 | M | D | D |
| 89 | 250842 | 78204093 | 830091 | GS373m | GeneralScience373m | 61 | GS373m4 | M | E | B |

The screen capture above indicates what's recorded when information for one candidate, 972311, ends and that for the following candidate, 830091, begins. In this case, each candidate was presented with precisely the same series of questions. The "Key" is again indicated in column 11 for each multiple-choice question – these do not change from one candidate to another as all candidates responded to exactly the same questions.

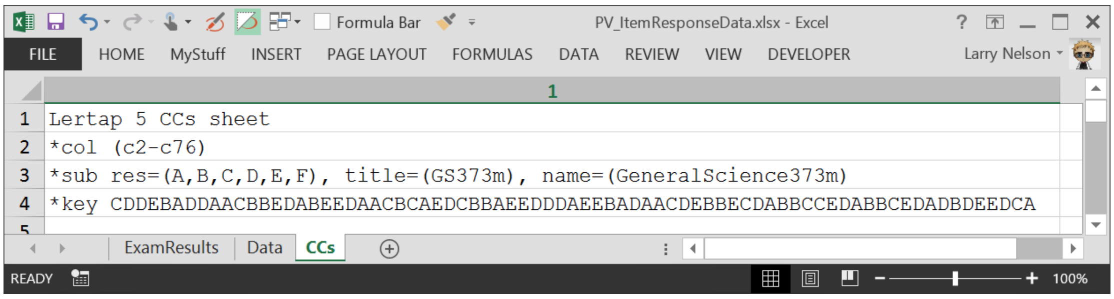
Note that the data in column 8 have changed. ClientCandidateID 972311 had a Score of 64, meaning the s/he got 64 of the 75 exam items correct. The following candidate did not do quite as well as her/his Score was 61.

## Applying the macro

The VUE spreadsheet has its data formatted in a manner quite different to that expected by Lertap 5. A macro was needed (and written) to take the VUE data apart and reformat it so that Lertap could be used.



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 77 | 78 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Item response data for GeneralScience373m | | | | | | | | | | | | | | | | | | | | |
| 2 | ClientCandidateID | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Q13 | Q14 | Q15 | Q16 | Q17 | Q18 | Score | |
| 3 | 972311 | A | D | D | B | B | A | C | D | A | A | C | B | B | E | E | A | B | E | 64 | |
| 4 | 830091 | B | C | D | B | B | A | C | D | B | A | C | B | B | E | E | A | B | E | 61 | |
| 5 | 734977 | A | E | A | B | D | A | C | B | B | E | C | B | D | A | E | B | C | A | 59 | |
| 6 | 335532 | B | C | B | B | B | A | A | D | C | D | C | E | B | E | A | B | B | B | 58 | |
| 7 | 1987965 | B | C | A | B | E | A | C | D | B | A | E | B | B | E | E | A | A | C | 61 | |



```
Lertap 5 CCs sheet
*col (c2-c76)
*sub res=(A,B,C,D,E,F), title=(GS373m), name=(GeneralScience373m)
*key CDDEBADDAACBBEDABEEDAACBCAEDCBBAEEDDDAEEBADAACDEBBECDABBCCEDABBCEDADBDEEDCA
```

The screen snapshots above indicate what happens when the macro runs. It creates the Data and CCs worksheets required by Lertap 5[2].

All may look pretty spiffy in these new sheets, Data and CCs, but there is at least one limitation: the `res=(A,B,C,D,E,F)` line in the CCs sheet is only a "stab"; the macro has not really determined the true nature of the response codes used in the test – it has simply left a "place holder"; users have to fix it so that it's correct. (In this case the items used {A,B,C,D,E} as response codes, but the macro has not been programmed to detect this – the user will have to change `res=(A,B,C,D,E,F)` to `res=(A,B,C,D,E)`.)

Experience Lertap 5 users may wonder why I have programmed the macro to carry over the candidate's test score, that column called "Score" by VUE.

The old Scantron utilities used to include the same information in their output files, and I recommend carrying it along as it can be used to test Lertap's scoring. How would that be done? Pretty easily: after Lertap's Interpret and Elmillon options have completed their tasks, the test score found in Lertap 5's "Scores" worksheet should equal the candidate "Score" found in the VUE worksheet.

If I copy the VUE score from the Data sheet over to the Scores worksheet, I should find a perfect correlation between the scores. There's an option on Lertap 5's "Move+" menu which will do this, and you may want to read about it here. It takes all of about 10 seconds to complete this useful little crosscheck.

## About the macro

The language commonly used to write macros in Microsoft Office apps is **VBA**, Visual Basic for Applications. There's an Excel workbook made exclusively to host VBA macros for use with Lertap 5. It's called "**Lertap5MacroSetA.xlam**". Whenever Lertap 5 starts, this macro-only Excel workbook is automatically loaded into memory and made available for use. All of the VBA code in the workbook is open, ready for modification at any time by anyone. Thus far I have authored almost all of the macros in Lertap5MacroSetA; one or two have been contributed by others.

For backgrounding on the use of the macros in Lertap5MacroSetA.xlam and an explanation of how Lertap is able to call on their services, I recommend starting with a study of the material in this webpage. You'll see that the tool used to create and edit macros is the "**Visual Basic Editor**". The way this editor is accessed differs depending on the version of Excel in use. I suggest a search engine could be asked "How to access the **VBE** in Excel 2010", for example, to find out how to start it with Excel 2010.

In Lertap5MacroSetA.xlam, the name of the VBA code module designed for working with VUE Excel worksheets is named "`PearsonVUE`". Within this module, the subroutine labelled "PVueExamSeries1()" is responsible for initiating the actions described in this document.

The purpose of the macro is, as mentioned above, to reformat the original data as found in the ExamResults worksheet so that Lertap will be able to analyse the candidates' item responses. Two new worksheets, Data and CCs, will be added to the workbook by the macro, and then filled out. The Data sheet will have a row

[2] Many of the columns in the Data sheet have been hidden in order to be able to display the Score column.

of results for each candidate, including ID information at the start of the row, followed by the candidate's item responses, one column for each response.

The CCs worksheet will need a *col line to tell Lertap and Excel where the item response columns are in the Data worksheet. It will need a *key line with the correct answers for the items. And it will be good to include a *sub line with an Res=( ) setting to remind the user of the need to define the response codes used by the test's items. (Lertap's default setting is Res=(A,B,C,D); in our present example Res=(A,B,C,D,E is what will be needed.)

Let's think for a moment about two of the CCs lines which will be required: *col, and *key. I'll show you bits of VBA code from the macro to exemplify how parts of the macro were written and functioned.

The purpose of the *col line is to tell Lertap and Excel where item responses are found. When you looked closely at the *col line in one of the screenshots above, you will have seen that the macro created `*col (c2-c76)`, meaning that candidate item responses start in Data worksheet column 2 and end 75 columns later in column 76.

How did the macro determine this?

Well, first of all it needed to know two things: how many ID fields were to be used to identify each candidate, and how many items were in the test.

Of these two "things", in this case the number of items in the test can be determined by having the macro read down the "Key" column until there aren't any more entries. The keys start in row 4 of the ExamResults worksheet and end in row 79. This you can see by looking at the first two screen snapshots above, at the start of this document.

The section of macro code below will form a string called "`HiloDeClaves`" which will contain the item keys, and then it will be able to set "`Nits`", the number of items, by getting the length of this string:

```
'The number of items, and the keys (correct answers), can be found by
reading down the 'Key' column for the first candidate until an empty cell is
found:

Do
  Let i = i + 1
  Let VUEcorrectItemResponse=
    Worksheets(VUEworksheetName).Cells(i,ColumnNumberForCorrectItemResponse)
  If Len(VUEcorrectItemResponse) > 1 Then
    MsgBox "Found a 'correct item response' in the VUE file longer than one
     character (this is unexpected; " & _
     "only the first character will be used in Lertap's *key CCs line)."
    Let VUEcorrectItemResponse = Left(VUEcorrectItemResponse, 1)
  End If
  Let HiloDeClaves = HiloDeClaves + VUEcorrectItemResponse
Loop Until IsEmpty(Worksheets(VUEworksheetName).Cells(i,
  ColumnNumberForCorrectItemResponse))

Let Nits = Len(HiloDeClaves) 'The number of items will equal the length of
the string of item keys (the correct answers)
```

Prior to starting to execute this code, the variable "`i`" was set to a value of 3, and `ColumnNumberForCorrectItemResponse` was set to a value of 11.

Note that this section of the macro makes a check on the length of each correct answer: `If Len(VUEcorrectItemResponse) > 1 Then` . The purpose of this check is to keep things "honest": the key for each item should be just a single character in this case (there are times with VUE datasets when the key to an item can be longer than one character, but this is not one of those "times"). Should an entry longer than one character be found for an item's key, the macro displays a message with a warning by using the `MsgBox` statement. Such statements cause the macro to pause while the message is shown to the user.

Also note that the `Do` loop uses an `Until IsEmpty` statement to terminate and exit. It's able to do this as we know in advance that each candidate's answers to the multiple-choice questions (those with an "M" in column 10) are followed by a small set of demographic questions, and these do not have a key, that is, do not have an entry in column 11. So, an empty key means we've come to the end of the multiple-choice items.

Okay – so much for determining the number of items ("Nits") and the string of item keys ("HiloDeClaves"). We're almost set to write the *col line, but first we have to know how many ID fields we are to take from the ExamResults worksheet for use in the newly-created Data worksheet.

The following code sets this up:

```
'Define location of the ID fields, as found in the VUE worksheet
    Let NumberOfIDfields = 1
    ReDim ColumnNumberForIDfield(NumberOfIDfields)
      Let ColumnNumberForIDfield(1) = 3 'This is to be the primary ID field
      'Let ColumnNumberForIDfield(2) = 2 'Could be CandidateID
      'Let ColumnNumberForIDfield(3) = 1 'Could be RegistrationID
```

Here we're telling the macro that there will be just a single ID field, the one found in column 3 of the ExamResults file. (The code lines which begin with a single quotation mark are comments, ignored by the macro.)

Right. Now we can get the macro to write the *col line, `*col (c2-c76)`. Easy.

(This assumes the single ID field will be written in the first column of each Data row, and that it is safe to assume that item responses are to follow immediately, starting in column 2, extending to column 76 since there are 75 items.)

And we can now get the macro to write the *key line too. Excessively easy.

```
'Write the *key line in the row 4 of the CCs sheet
   Worksheets("CCs").Cells(4, 1) = "*key " + HiloDeClaves
```

Are you catching on, isn't this coding stuff simple and straightforward?

Actually for experienced programmers it may be exactly that. VBA is not a difficult language. It's backed up by extensive documentation and all sorts of supporting websites.

Now, even if you do not open up the VBE to look at the macro code, you should be aware of this: the macro can fail.

The macros in this example make a number of assumptions: the length of each item key is just a single character; the length of each candidate response is also just a single character; each candidate has some "extra" item "responses" after his or her responses to the multiple-choice items, and these items do not have an entry in the key column, allowing the macro to determine the number of multiple-choice items; each and every candidate has the same number of rows in the ExamResults worksheet: they all were asked to respond to the same number of multiple-choice items, and also to the same number of "extra" items; each and every candidate has information in the primary ID field in the ExamResults worksheet, the field which gets copied over to the rows in the Data worksheet (this is important as that information will copy into column 1 of the Data sheet, and Lertap thinks that it's come to the end of data whenever it encounters and empty cell in the first column of a Data row).

## Adapting the macro

If you get into the Lertap5MacroSetA.xlam file for a look at the "`PearsonVUE`" module, you'll notice that it contains extensive commenting, the idea being that it can be locally adapted, and expanded as needed.

The little "ExamResults" worksheet used in this example is typical of VUE worksheets in the way fields are named and formatted. Often there will be many more fields; for example, VUE CID documentation sets out no fewer than 25 fields just for candidate demographic data. In my example, I have used just two of these: CandidateID and ClientCandidateID. Usually there will be more, such as FirstName, LastName, MiddleName, Suffix (such as "Jr."), Salutation ("Mrs., "Mr.", and so on) and a veritable host of address fields. All of these may be "accommodated" in the macro – just increase the `NumberOfIDfields` setting above, and change the corresponding column designations.

VUE will in some cases also supply "Item Results" information when, for example, item scoring is more complex than the simple (and common) scoring seen in this document, where each item has a single scored answer. Lertap has always allowed for very complex item scoring; I haven't seen anything yet in VUE which could not be done in Lertap, but things can get a bit hairy; I might have to change this comment someday.

It would be very easy to envisage a situation where the multiple-choice item responses in a VUE worksheet are not followed by a few "extra" item responses, as found in the example above. This would certainly affect my macro, forcing it to find another way of determining the number of items and the string of item keys – above, as mentioned, the present macro looks for a break in the item key column (column 11 above), but there wouldn't be one if there weren't any "extra" items. What I might do in that case is get the ItemName of the first item, and then cruise down the column for ItemName (column 9 above) until the name of the first item is found again. (Above that happens in row 86 of the ExamResults worksheet.)

## Help with macros

Feel very free to write us if you might want more information about macros and Lertap. In some cases we'll write macros for free while in other cases we can often be tempted into writing special ones on a fee basis.

Cheers!

---

Larry Nelson (support@lertap.com)