

# Concerto v3 Tutorials: Tests with Likert-type rating scales

This tutorial will show you how to create a test that has multiple response options (likert-type rating scale) on Concerto v3. It is assumed that you've already taken the **Concerto v3.6 Tutorials: Create a simple test** (<http://code.google.com/p/concerto-platform/wiki/Concerto3SimpleTest>) and that you are familiar with the basic functions of Concerto v3.

## Login to Concerto

It is best to use Google Chrome to run the Concerto administration panel. Please ensure that you have access to the latest version (i.e. Concerto v3.6). Next, go to **concerto\_installation\_path/cms/** and login using your credentials. If you haven't installed Concerto on your server, you can use our demo installation.

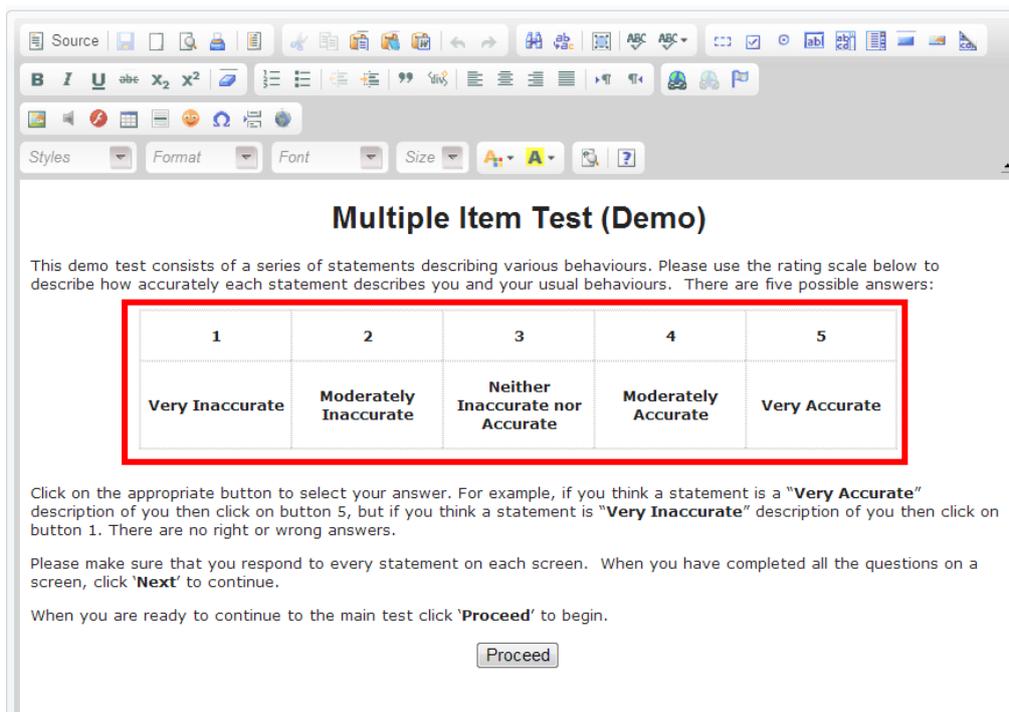
Go to the demo installation (<http://concerto.e-psychometrics.com/demo/cms/index.php>) and click "Register" to create a demo account for yourself.

## Step 1: Create HTML Templates

### Introduction templates

Create a new template called *Multi Item\_1* (or whatever you prefer):

- add some **suitable text** to describe the test and make participants feel comfortable.
- Make sure that you also introduce the **scale** used (see red box).
- Also add a **button** (called "Proceed", for example) at the bottom of the page – to proceed to the next page.



**Multiple Item Test (Demo)**

This demo test consists of a series of statements describing various behaviours. Please use the rating scale below to describe how accurately each statement describes you and your usual behaviours. There are five possible answers:

1	2	3	4	5
Very Inaccurate	Moderately Inaccurate	Neither Inaccurate nor Accurate	Moderately Accurate	Very Accurate

Click on the appropriate button to select your answer. For example, if you think a statement is a "Very Accurate" description of you then click on button 5, but if you think a statement is "Very Inaccurate" description of you then click on button 1. There are no right or wrong answers.

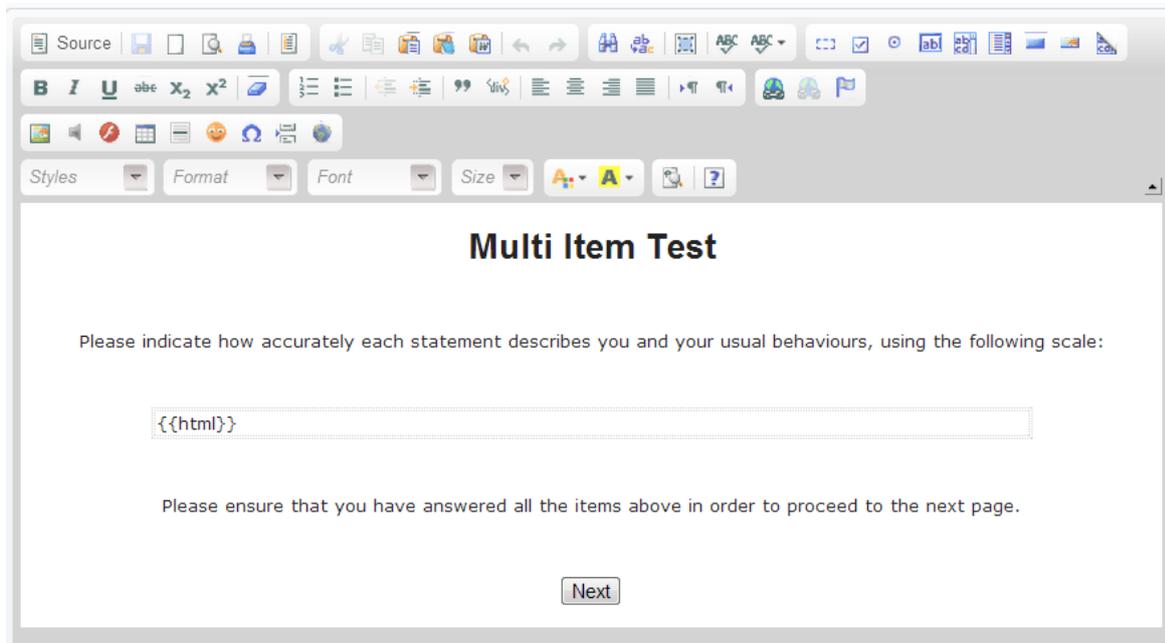
Please make sure that you respond to every statement on each screen. When you have completed all the questions on a screen, click 'Next' to continue.

When you are ready to continue to the main test click 'Proceed' to begin.

## Test item template

Create a new template called *Multi Item\_2* and add the following content:

- **{{html}}** insert that will be filled by concerto
- A button called "Next"
- Any text (instructions etc.) and formatting that you wish.



The screenshot shows a web editor interface with a toolbar at the top. The main content area contains the following text:

### Multi Item Test

Please indicate how accurately each statement describes you and your usual behaviours, using the following scale:

Placeholder: `{{html}}`

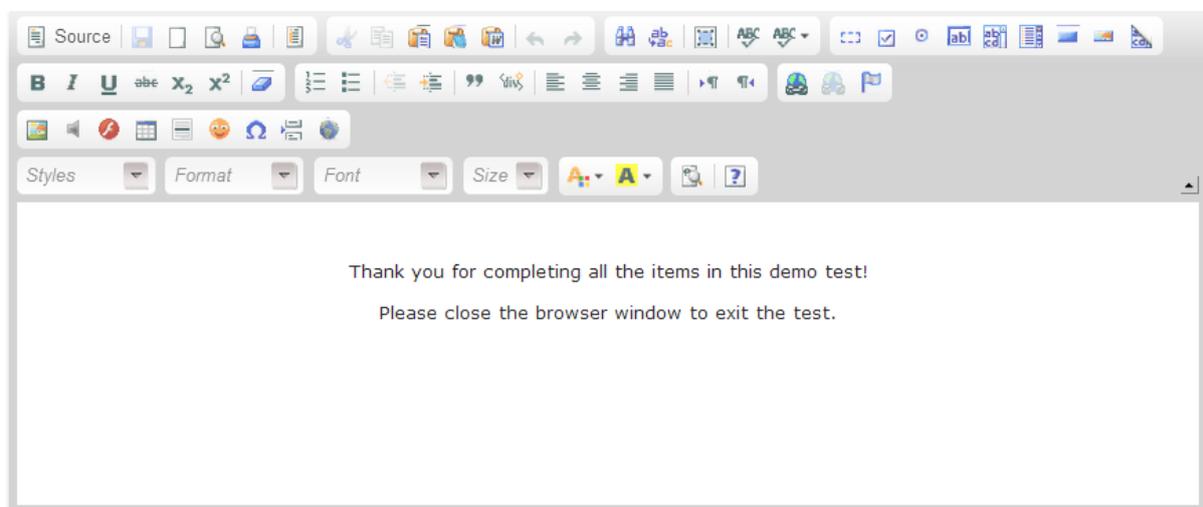
Please ensure that you have answered all the items above in order to proceed to the next page.

Next

## Feedback template

Create a third new template called *Multi Item\_3* that contains:

- Suitable text and formatting, including a **thank you** note for participation and few words of empathic feedback, if necessary.



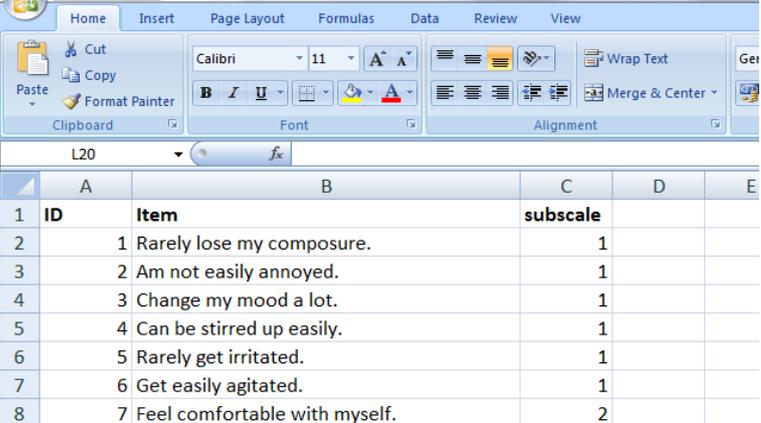
The screenshot shows a web editor interface with a toolbar at the top. The main content area contains the following text:

Thank you for completing all the items in this demo test!

Please close the browser window to exit the test.

## Step 2: Creating an Item Bank/Table from a CSV File

The next step is to create an **item bank** for the test. You can either directly type in the items into a table (this is not the preferred option when you have more than a handful of items), or *import a csv file* that contains the items needed for the test. This sample test uses a csv file (create an excel table and save as .csv format).



	A	B	C	D	E
1	<b>ID</b>	<b>Item</b>	<b>subscales</b>		
2	1	Rarely lose my composure.	1		
3	2	Am not easily annoyed.	1		
4	3	Change my mood a lot.	1		
5	4	Can be stirred up easily.	1		
6	5	Rarely get irritated.	1		
7	6	Get easily agitated.	1		
8	7	Feel comfortable with myself.	2		

- Go to the **tables** tab and click the '+ add new object' icon to add a new table.
- Assign a meaningful name (E.g. *Multi Item\_items*) and save it.
- If you have the items in a csv file, click the *import table from CSV file* icon to import a table that you have downloaded. Concerto will automatically read the data into table format. All you need to do is alter the column names and the data types (see red box).

editing table #55

\* name: Multi Item\_items

table structure definition

+ add

name	type	length	default	attributes	nullable	auto increment
ID				int		
Item				text		
subscales				int		

table data

+ add clear table

ID	Item	subscales
1	Rarely lose my composure.	1
2	Am not easily annoyed.	1
3	Change my mood a lot.	1
4	Can be stirred up easily.	1
5	Rarely get irritated.	1

### Step 3: Creating a table to store user data

The screenshot shows the 'editing table #56' interface. At the top, the name is 'Multi Item\_user data'. Below is the 'table structure definition' section, which contains a table with the following columns and rows:

name	type	
test_id	int	[edit] [delete]
session_id	int	[edit] [delete]
item_no	int	[edit] [delete]
response	int	[edit] [delete]

Below the structure definition is the 'table data' section, which contains a table with the following columns and rows:

test_id	session_id	item_no	response	
33	3333	1	0	[delete]
33	3333	2	0	[delete]

Now you need to create a table which is suitable to store the **user responses**.

- Go to the **tables** tab and click the '+ add new object' icon to add a new table.
- Assign a meaningful name (E.g. *Multi Item\_user data*) and save it.
- Insert variable names and *specify the data type* of the variable (see red box). In the simple view, you can select between numeric and text. In this picture, we have used the advanced options to select 'int'. *Int* means integer and is a numerical value, but you can of course also have text values (*text*).
- To change the variables, either press the "+add" button to create a new item or edit existing variables by clicking on the little pencil (green boxes).
- The table data section below will be inserted automatically by Concerto. If you have no user responses, it will be empty.

### Step 4: Creating a test and adding sections

In the **'tests'** tab, create a new test by adding a new object. Follow the images and corresponding text to create the structure of your test by adding new sections.

**Annotation:** The numbers of the sections are automatically given by Concerto and will change depending on when you add or delete sections. So your test will have differing numbers but this does not change the functionality of the test!

Each new section can be added with the **"+add section"** button (green box). Then you have to choose the type of section (e.g. set variable, load HTML template...). The right section for each step is described in the SECTION TYPE column of the tables.

? test logic

---

1 ? start + add section

**returns**

- ? CONCERTO\_TEST\_ID
- ? CONCERTO\_TEST\_SESSION\_ID

---

68 ? R code + add section

```

1 for(i in 1:40){
2   assign(paste("response", i, sep=''), NA, envir=.GlobalEnv)
3 }

```

---

3 ? load HTML template + add section

Multi Item\_1 ( id: 86, owner: Vaishali Mahalingam )

parameters:	returns:
<ul style="list-style-type: none"> <li>? TIME_LIMIT</li> </ul>	<ul style="list-style-type: none"> <li>? LAST_PRESSED_BUTTON_NAME</li> <li>? OUT_OF_TIME</li> <li>? TIME_TAKEN</li> </ul>

---

31 ? set variable + add section

count

from table  , by R code

1 1

Figure 1

SEC-TION TYPE	DETAILS
1: start	-This is automatically inserted by Concerto
68: R code	<p>-Please copy the text accurately. It creates a “for loop” which is important for storing the responses of the participants later. You create 40 variables (response1, response2....response40) and assign values of NA for all. If a person does not select an answer (via the radio buttons on the items template) Concerto will recognise that this variable still exists. Without this loop Concerto would not be able to locate the response variables since radio buttons are non-existent variables until they are selected.</p> <p>-This is the code:</p> <pre>for(i in 1:40) {   assign(paste("response",i,sep=""),NA,envir=.GlobalEnv) }</pre>
3: load HTML template	-Select from the drop-down menu the “ <b>Multi Item_1</b> ” template you created in Step 1 (introduction)
31: set variable	<p>-Create a new variable called “<b>count</b>” (by typing in the box)</p> <p>-select by <b>R code</b></p> <p>-set value to <b>1</b></p>
<b>See Figure 1</b>	

35 ? set variable    
 from table , by R code   
 1 ""

46 ? set variable    
 from table , by R code   
 1 1

29 ? set variable    
 from table , by R code   
 select table: + ?   
 select columns:    
 where:

50 ? set variable    
 from table , by R code   
 1 "<div style='text-align: right;'\>  
 2 <input name='response' type='radio' value='1' /><input name='response' type='radio' value='2' /><input name='response' type='radio'  
 value='3' /><input name='response' type='radio' value='4' /><input name='response' type='radio' value='5' /></div>"

Figure 2

SECTION TYPE	DETAILS
35: set variable	-Create a new variable called " <i>html</i> " -select by <b>R code</b> -set value to "" (empty double quotes)
46: set variable	-Create a new variable called " <i>total_count</i> " -select by <b>R code</b> -set value to 1
29: set variable	-Create a new variable called " <i>question</i> " -select <b>from table</b> option -choose the table " <i>Multi Item_items</i> " (created in Step 2) from the drop-down menu -select columns: <b>Item</b> -where: <b>ID equal count</b>
50: set variable	-Create a new variable called " <i>likert_scale</i> " -select by <b>R code</b> -set value to: " <code>&lt;div style='text-align: right; '&gt;                &lt;input name='response' type='radio' value='1' /&gt;&lt;input name='response' type='radio' value='2' /&gt;&lt;input                name='response' type='radio' value='3' /&gt;&lt;input name='response' type='radio' value='4' /&gt;&lt;input name='response' type='radio'                value='5' /&gt;&lt;/div&gt;</code> "  Make sure that you copy this HTML code exactly. It produces a Likert-Scale with 5 response options.
<b>See Figure 2</b>	

55 ? R code +add section

```
1 likert_scale<- gsub("response", paste("response", count, sep=""), likert_scale)
```

---

67 ? set variable  +add section

from table , by R code

```
1 paste(html, "<table width='100%><tr><td style='text-align: left;'", question, "</td><td style='text-align: right;'", likert_scale, "
</td></tr></table>", sep="")
```

---

32 ? set variable  +add section

from table , by R code

```
1 count+1
```

---

38 ? IF statement    +add section

additional conditions:

AND    -

+

+add nested section

39 ? go to  +add section

Figure 3

SECTION TYPE	DETAILS
55: R Code	-Copy the R code exactly. It substitutes the name of the likert scale radio buttons with the number of the item (i.e. response1... response40).  <b>likert_scale&lt;&lt;- gsub("response", paste("response", count, sep=""), likert_scale)</b>
67: set variable	-Now you specify the variable " <b>html</b> " you created before. This code pastes the 'question' and 'likert_scale' together, along with some formatting (HTML code) to present the participant with a full item. Further, this variable is part of a loop that will paste 5 such items together in order to present five items on a page. - <b>by R code</b> - code: <b>paste(html, "&lt;table width='100%'&gt;&lt;tr&gt;&lt;td style='text-align: left;'&gt;", question, "&lt;/td&gt;&lt;td style='text-align: right;'&gt;", likert_scale, "&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;", sep="")</b>
32: set variable	-Now you modify the variable " <b>count</b> " you have created earlier -select by <b>R code</b> -set it to <b>count+1</b> (so right now it will have a value of 2, the next time 3 and so on)
38: IF statement	-Now you specify the stopping rule for the number of items on each page and the total items in the test. -please do it exactly like on the screenshot: if <b>count equal or lesser than</b> (choose from the drop-down menu) <b>total_count+9</b> (this step ensures that there are 10 items on a page) - <b>additional conditions: AND count equal or lesser than 40</b> (this step ensures that Concerto knows when to stop presenting items and go to the feedback page) -Select " <b>+add a nested section</b> " (see red box) which is <i>indented</i>
NESTED SECTION 39: go to	-This section makes sure that if the demands from the IF statement are met Concerto loops to a certain point in the test, in this case to section 29 where you defined the variable " <b>question</b> " -select from the drop-down menu <b>the section where you defined the variable "question"</b> (it will have a different section number in your test!) -only use the add section button within the red box, if you want to insert another indented section! Otherwise use the " <b>+add section</b> " button from the <b>IF statement section above</b> .
<b>See Figure 3</b>	

37 ? load HTML template Multi Item\_2 ( id: 87, owner: Vaishali Mahalingam ) + add section

parameters:	returns:
? html	? LAST_PRESSED_BUTTON_NAME
? TIME_LIMIT	? OUT_OF_TIME
	? TIME_TAKEN

48 ? set variable responses + add section

from table , by R code

```
1 matrix(c(count-10, count-9, count-8, count-7, count-6, count-5, count-4, count-3, count-2, count-1, get(paste("response",
(count-10), sep="")), get(paste("response", (count-9), sep="")), get(paste("response", (count-8), sep="")), get(paste("response",
(count-7), sep="")), get(paste("response", (count-6), sep="")), get(paste("response", (count-5), sep="")), get(paste("response",
(count-4), sep="")), get(paste("response", (count-3), sep="")), get(paste("response", (count-2), sep="")), get(paste("response",
(count-1), sep="))), nrow=10)
```

23 ? set variable insert\_index + add section

from table , by R code

```
1 1
```

Figure 4

SECTION TYPE	DETAILS
37: load HTML template	-Select from the drop-down menu the “ <i>Multi Item_2</i> ” template you created in Step 1 (test item template)
48: set variable	<p>-Create a new variable called “<i>responses</i>”            -select by <b>R code</b>            -set it to:</p> <pre>matrix(c(count-10,count-9,count-8,count-7,count-6,count-5,count-4,count-3,count-2,count-1,get(paste("response",(count-10),sep="")),get(paste("response",(count-9),sep="")),get(paste("response",(count-8),sep="")),get(paste("response",(count-7),sep="")),get(paste("response",(count-6),sep="")),get(paste("response",(count-5),sep="")),get(paste("response",(count-4),sep="")),get(paste("response",(count-3),sep="")),get(paste("response",(count-2),sep="")),get(paste("response",(count-1),sep="))), nrow=10)</pre> <p>-It creates a matrix to temporarily store item numbers and responses from participants, which will then be transferred row by row to the user response table. Make sure that you copy the code exactly.</p>
23: set variable	<p>-create a new variable called “<i>insert_index</i>”            -select by <b>R code</b>            -set it to <b>1</b></p>
<b>See Figure 4</b>	

56 ? R code + add section

```
1 item<-responses[insert_index,1]
2 resp<-responses[insert_index,2]
```

---

24 ? table modification Multi Item\_user data ( id: 56, owner: Vaishali Mahalingam ) + add section

insert , update , delete

set:

test\_id  -

session\_id  -

item\_no  -

response  -

+

---

57 ? set variable insert\_index + add section

from table , by R code

```
1 insert_index+1
```

Figure 5

SECTION TYPE	DETAILS
56: R code	<p>-Type in two lines:</p> <pre><b>item&lt;-responses[insert_index,1]</b> <b>resp&lt;-responses[insert_index,2]</b></pre> <p>-this creates two new variables, which only use one row and one column of the “<i>responses</i>” matrix you have created before. We use the <i>insert_index</i> variable as a counter to draw one line at a time from the <i>responses</i> matrix. <i>item</i> variable corresponds to the first column while <i>resp</i> variable corresponds to the second column, respectively.</p>
24: table modification	<p>-choose the table “<b><i>Multi Item_user data</i></b>” you have created in Step 3          -choose radio button: <b>insert</b>          - Set:  <b>test_id = CONCERTO_TEST ID</b>  <b>session_id = CONCERTO_TEST_SESSION_ID</b>  <b>item_no = item</b>  <b>response = resp</b></p>
57: set variable	<p>-Now you modify the variable “<b><i>insert_index</i></b>” you have created before such that it increases by 1.          -select by <b>R code</b>          -set it to <b>insert_index+1</b> (so right now it will have a value of 2, the next time 3 and so on)</p>
<b>See Figure 5</b>	

58 ? IF statement if insert\_index equal or lesser than 10 + add section

additional conditions:  
+

{ + add nested section

59 ? go to 56: R code + add section

}

47 ? set variable html + add section

from table , by R code

1 ""

45 ? set variable total\_count + add section

from table , by R code

1 total\_count+10

Figure 6

SECTION TYPE	DETAILS
58: IF statement	<p>-Now you specify under which circumstances the test should loop back to a certain point            -please do it exactly like on the screenshot:            if <b><i>insert_index</i></b> - <b>equal or lesser than</b> (choose from the drop-down menu) - <b>10</b>            -press the <b><i>+add nested section</i></b> which then should be indented (see red box)</p>
NESTED SECTION 59: go to	<p>-This makes sure that if the demands from the IF statement section are met it goes back to a certain point, in this case to section 56 where you defined the variables <i>item</i> and <i>resp</i> by giving them R codes            -select from the drop-down menu <b>the section where you defined the variables <i>item</i> and <i>resp</i> by R code</b>, which will have a different number in your test.            -only use the add section button within the red box, if you want to insert another indented section! Otherwise use the +add section button from the IF statement section to create a test section outside of the IF statement, as you were doing before.</p>
47: set variable	<p>-Reset the variable <i>htm</i> you have created before, so that fresh <i>question</i> and <i>likert_scale</i> variables can be pasted together to be presented to the participant.            -select by <b>R code</b>            -set it to ""</p>
45: set variable	<p>-Now you modify the variable <b><i>total_count</i></b> you have created before            -select by <b>R code</b>            -set it to <b><i>total_count+10</i></b> (so right now it will have a value of 11, the next time 21 and so on. This variable is used as a page counter)</p>
<b>See Figure 6</b>	

43 ? IF statement if count equal or lesser than 40 + add section

additional conditions:

+ add nested section

44 ? go to 29: set variable + add section

16 ? load HTML template Multi Item\_3 ( id: 88, owner: Vaishali Mahalingam ) + add section

parameters:

- ? TIME\_LIMIT

returns:

- ? LAST\_PRESSED\_BUTTON\_NAME
- ? OUT\_OF\_TIME
- ? TIME\_TAKEN

2 ? end

Figure 7

SECTION TYPE	DETAILS
43: IF statement	-Now you specify under which circumstances the presentation of items should end (stopping rule for the test). -please do it exactly like on the screenshot: if <b>count - equal or lesser than</b> (choose from the drop-down menu) - <b>40</b> -add nested section which is indented (see red box)
NESTED SECTION 44: go to	-This makes sure that if the demands from the IF statement section are met, a loop is created back to a certain point – in this case to section 29 where you defined the variable “ <i>question</i> ”; if not, the final feedback template (next section) will be presented. -select from the drop-down menu the section where you defined the variable “ <i>question</i> ”, which surely has a different number in your test -only use the add section button within the red box, if you want to insert another indented section! Otherwise use the +add section button from the IF statement section
16: load HTML template	Select from the drop-down menu the template “ <b>Multi Item_3</b> ” you created in Step 1 (feedback)
2:end	This section is automatically inserted by Concerto
<b>See Figure 7</b>	
<b>END section</b>	

## Run your test

Use the following URL to run your test: **concerto\_installation\_path/?tid=TEST\_ID**.

For the demo installation, the link is: <http://concerto.e-psychometrics.com/demo/?tid=XX>

*Note: Substitute the last 2 digits in the URL with the test id of your choice*