Contents

XML

Settings Features: Add different features to the survey that do not appear by default.

Items Features: Manipulate how questions appear on the survey.

Filtering Features: Show or hide questions on the survey for different scenarios.

Branching Features: Redirect the user to a specific page or question when a criteria is met.

6. Branching using event assignments
   🕮 Branch from item 20 to item 50 if somebody uses one of the two event assignments.
7. Take users to a specific item regardless of response
   🕮 Use Case: Anyone who gets to item 2 will go to Q121 regardless of what they answered.
8. Conditional sets
   🕮 Useful when you have a condition that's used multiple times throughout the XML and use a label to identify it.
9. Conditional redirection
   🕮 Use Case: Custom Finish Text commonly used for kiosk. The page goes to the finishing page which could be the login page for the next user to take the survey.
10. Conditional finish page text
    🕮 Use Case: Determine what the user see's based on if they qualify for an individual engagement report. Send the User to a Thank you page, then to their report.
11. Survey finish time
    🕮 Use Case: Checks if the respondent completed the survey in less than 120 seconds. If it's true it returns a custom finish text and/or has the ability to redirect to a special location

Conditional Features: Expand on existing feature and add specific criteria using execution statements.

1. Event Assignment ID
   🕮 Use case: when you want to use event assignment ID as part of your condition.
2. Item Response (can be tie back)
   🕮 Use case: when you want to use a response value as part of your condition.
3. Culture Code (language)
   🕮 Use case: when you want to use culture code as part of your condition.
4. Customer Value
   🕮 Use case: when you want to use custom value 1 or 2 as part of your condition.
5. Operator 'OR' Condition
   🕮 Use case: When you want to use OR as part of your condition.
6. Operator 'AND' Condition
   🕮 Use case: When you want to use AND as part of your condition.
7. Operator 'ANY' Condition
   🕮 Use case: When you want to use any condition.
8. Operator 'EXACT' Condition
   🕮 Use case: When you want to use EXACT as part of your condition.
9. Email Address Validation
   🕮 Use Case: Validate the email address has the specific criteria of an email address (@, .com, etc.) and return a message when incorrect.

Code Optimization: Examples of the best way to simplify code for different features.

1. Filtering response options
   🕮 When you have 5 response options, show the response option based on a condition.
2. Branching
   🕮 Combine the destination into one.
3. Auto response: Example 1
   🕮 Combine conditions when the target itemRef update is the same.
4. Auto response: Example 2
   🕮 Combine Conditions: For each unique Target Item Ref create one condition when the Target Item rows are the same.

XML

This page provides examples of XML codes for Research Survey configuration.

XML Basics

XML stands for Extensible Markup Language. It is a computer language very similar to HTML, it works by way of “tags”. They are ‘self closing’ or single tags but most tags ‘open’, surround something and ‘close’. Tags can have attributes as part of the opening tag. In Research many of the tags are our own - designed specifically for our software.

Tags appear in angle brackets like “<” and “>”. Here is an example of an HTML tag for making the word “Flower” appears in bold:

```
<b>Flower</b>
```

All closing tags contain a forward slash.

Here is the XML in Research to format a single item horizontally so that it appears similar to the table style of a “matrix” item.

```
<item itemRef="7">
 <settings>
  <renderAsBlock>1</renderAsBlock>
 </settings>
</item>
```

There are three tags in this sample; the ‘item’ tag (which has “itemref” as an attribute), the ‘settings’ tag and the ‘renderAsBlock’ tag. In order for this code to work for an m/survey in Research it must be in two more tags. All XML code must between the open and close of the ‘Survey’ tag. After that tags must be in one of the categorical tags; ‘settings’, ‘items’, ‘filtering’ or ‘branching’. You can put as many tags in the categorical tags as you want but they appear only once. So to use the above code you’d have to put this in Research:

```
<survey>
  <items>
    <item itemRef="7">
     <settings>
      <renderAsBlock>1</renderAsBlock>
     </settings>
    </item>
  </items>
</survey>
```

The ‘settings’ tag above is not the categorical tag ‘settings’ (which is for global survey settings) but is within the ‘item’ tag and relates to settings for the item referenced in that tag. In order to reduce confusion, tags in HTML and XML are indented to represent the level of the tag (each tag goes back one further tab). This convention does not affect the operation of the tags but is for readability and organization.

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>. Opening and closing tags must be written with the same case:

```
<RenderAsBlock>This is incorrect</renderasBlock>
<renderAsBlock >This is correct</renderAsBlock >
```

Thinking like a Programmer

Using XML in an m/survey is different from using some of our more user friendly interfaces like the survey designer of Research. Using XML will require that you think a bit more like a programmer. In order to help you think a little more like a programmer let's briefly discuss syntax, flexibility and quality control.

Syntax:

Computer languages are very unforgiving. Even a slight deviation, like an extra comma, can "break" the code. The exact "syntax" must be followed to get the desired outcome. If you write a lot of XML and your survey shows an error it may be difficult to identify the cause. The usual culprit will be syntax.

Flexibility:

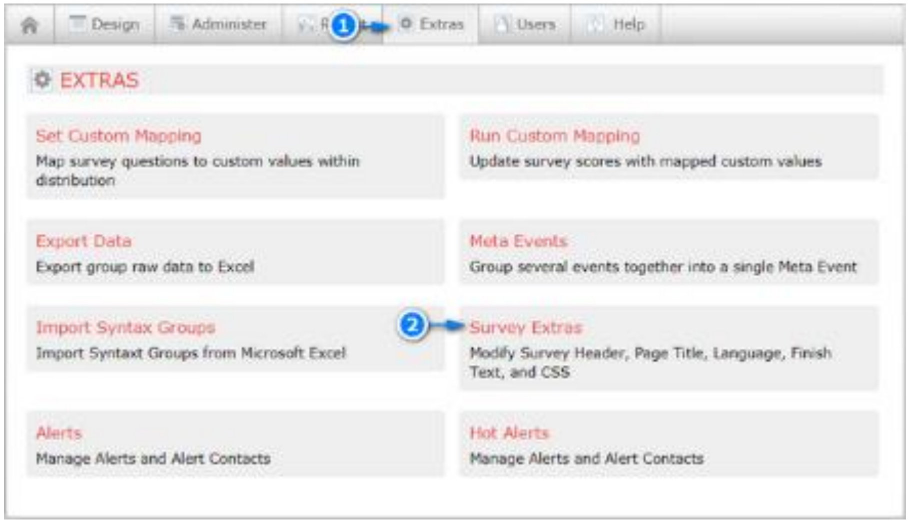
Usually when you use a program all the choices you can make have been tested and anything that wouldn't work or that hadn't been thoroughly tested wouldn't be available to you. Programming is different. You can easily find yourself using the XML for things we never thought of or in combinations we haven't tested. This code has been thoroughly tested for what the sample survey is doing but writing your own code includes the ability to pave new roads. We will of course help to navigate any new territory.

Quality Control:

QC is a part of every project but the kind of QC that is needed when you have the type of flexibility described above is different. The difference is that you have to think of ALL of the possible effects of your code including unintended effects to seemingly unrelated things. If, for instance, you use our new branching logic every possible branch should be checked including what happens if you skip the question and whether the skipped questions are part of the logic needed for anything else and so on.

Using XML in Research

XML is used on in Research in the Extras section to help program your survey to accommodate special formatting and functionality.
Navigate to Extras -> Survey Extras, you will find an Extended Definition XML area towards the bottom of the page. This is where you will enter all XML.



EXTRAS

Set Custom Mapping
Map survey questions to custom values within distribution

Run Custom Mapping
Update survey scores with mapped custom values

Export Data
Export group raw data to Excel

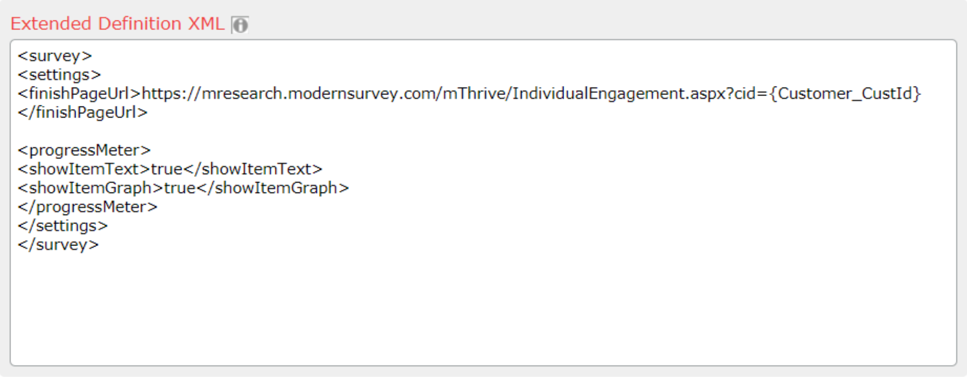
Meta Events
Group several events together into a single Meta Event

Import Syntax Groups
Import Syntaxt Groups from Microsoft Excel

Survey Extras
Modify Survey Header, Page Title, Language, Finish Text, and CSS

Alerts
Manage Alerts and Alert Contacts

Hot Alerts
Manage Alerts and Alert Contacts



Extended Definition XML

```
<survey>
<settings>
<finishPageUrl>https://mresearch.modernsurvey.com/mThrive/IndividualEngagement.aspx?cid={Customer_CustId}
</finishPageUrl>

<progressMeter>
<showItemText>true</showItemText>
<showItemGraph>true</showItemGraph>
</progressMeter>
</settings>
</survey>
```

XML Features and Codes

You cannot have multiple xml for an itemRef, the system will only accept the first one and ignore other instances.

Remember to add the survey open and close tags at the beginning and end of the entire XML. Do not add multiple times!

```
<survey>
  <settings>
Insert XML
  </settings>
  <items>
Insert XML
  </items>
  <branching>
Insert XML
  </branching>
  <filtering>
Insert XML
  </filtering>
</survey>
```

Here’s an example of XML with an explanation of the tags to help you understand the structure of including multiple tag types in one XML block.

<survey>	The survey starting tag. This should always appear at the beginning.
<settings>	The settings start tag. All XML for Settings will be between the settings tags. In this example there are two settings Clear Form Link and Progress Meter.
<survey>	The survey starting tag. Should always appear at the beginning.
<settings>	The settings start tag. All XML for Settings will be between the settings tags. In this example there are two settings Clear Form Link and Progress Meter.
<showClearFormLink>true</showClearFormLink>	The clear form XML
<progressMeter>	The progress meter starting tag
<showPagePercent>true</showPagePercent>	The progress meter XML
</progressMeter>	The progress meter closing tag
</settings>	The settings closing tag
<items>	The items start tag. All XML for items will be between the Items tags. In this example there are two question items being referenced: itemRef 1 and itemRef 5.
<item itemRef="1">	The item number identifier, this is the question number you want the XML to apply to.
<itemType>4</itemType>	The item XML
</item>	The item 1 closing Tag
<item itemRef="5">	The Item number identifier
<renderAsBlock>1</renderAsBlock>	The item XML
</item>	The item 5 closing tag
</items>	The items closing tag
</survey>	The survey closing tag. Should always appear at the end.

Use the tables below to navigate to the different XML features that we've prepared for you to get you started. All of these should appear between the identified tags. Remember to add the survey open <survey> and close </survey> tags!

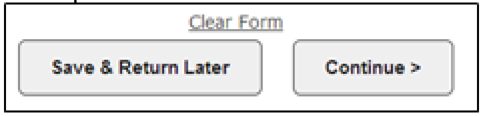
Settings Features

Settings features allows you to add different features to the survey that do not appear by default. These features will be between settings tags: <settings></settings>.

XML Example

1. Add a 'Clear Form' link at the bottom of the survey.

Example:



```
<settings>
<showClearFormLink>true</showClearFormLink>
</settings>
```

2. Customize the survey return button behavior

Default Behavior: They always are returned to the first page.

NoRedirect: The survey will behave as normal, starting at the first page of the survey upon return (default).
LastWithValuePage: Returns to the last page that has an item with a non-empty response from the user. Actual values submitted.
LastRespondedPage: Returns to the last page with a response from the user, whether or not it was empty. Could be they just saved empty responses. That happens when you navigate forward, backward or Save & Return a survey.
LastViewedPage: Returns to the last page the user saw, but didn't necessarily respond to. This determined by ViewStatus in the vertical scores page.

Last Viewed:

```
<settings>
<returnBehavior>LastViewedPage</returnBehavior>
</settings>
```

Last Responded:

```
<settings>
<returnBehavior>LastRespondedPage</returnBehavior>
</settings>
```

Last With Value Page:

```
<settings>
<returnBehavior>LastWithValuePage</returnBehavior>
</settings>
```

Survey Return Button Examples:

	NoRedirect	LastWithValuePage	LastRespondedPage	LastViewedPage
Open survey (splash page) -> Go to Page 1 and answer all questions - > go to page 2 – Save & Return	**First Page**	**Page 1**	**Page 2**	**Page 2**
Open survey (splash page) -> Go to Page 1 and answer all questions - > go to page 2 – Close the browser	**First Page**	**Page 1**	**Page 1**	**Page 2**
Open survey (splash page) - > Answer on page 1 -> click through page 2 - > Click through page 3 - > go to page 4 -> Save & Return	**First Page**	**Page 1**	**Page 4**	**Page 4**
Open survey (splash page) - > Answer on page 1 -> click through page 2 - > Click through page 3 - > go to page 4 -> Close browser	**First Page**	**Page 1**	**Page 3**	**Page 4**
Open survey (splash page) - > Answer on page 1 -> click through page 2 - > Click through page 3 - > go back to page 2 -> Save & Return	**First Page**	**Page 1**	**Page 3**	**Page 3**

Settings Features

Settings features allows you to add different features to the survey that do not appear by default. These features will be between settings tags: <settings></settings>.

XML Example

3. Add questions to an ongoing survey

Use Case: When modifying a live survey, add questions to the survey designer and insert them where you want them to display. This prevents re-numbering of items for users who have already taken the survey.

1. Add questions to the end of the survey in the Survey Designer then slot them into a different order on the survey. The following is an example of questions 104, 109, and 115 inserted into the middle of the survey.
2. Stop sequencing when you get to the item numbers you want inserted.

```
<settings>
<itemSequence>
<item sequence="1" itemRef="1" />
<item sequence="2" itemRef="2" />
<item sequence="3" itemRef="3" />
<item sequence="4" itemRef="4" />
<item sequence="5" itemRef="104" />
<item sequence="6" itemRef="109" />
<item sequence="7" itemRef="115" />
<item sequence="8" itemRef="5" />
</itemSequence>
</settings>
```

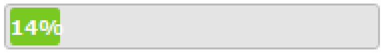
4. Add a progress meter

Add a Progress Meter at the top of the page, **only two meters are supported** in a survey header.

Use any of the options below based on your desired results.

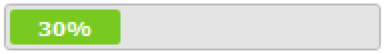
showItemText = Item 8 of 23

showItemPercent = 14% Complete

showItemGraph = 

showPageText = Page 2 of 7

showPagePercent = 30% Complete

showPageGraph = 

```
<settings>
<progressMeter>
<showItemText>true</showItemText>
<showItemPercent>true</showItemPercent>
<showItemGraph>true</showItemGraph>
<showPageText>true</showPageText>
<showPagePercent>true</showPagePercent>
<showPageGraph>true</showPageGraph>
</progressMeter>
</settings>
```

Settings Features Settings features allows you to add different features to the survey that do not appear by default. These features will be between settings tags: <settings></settings>.	**XML Example**

5. Display dynamic logos
Use Case: Dynamically display a logo based on a condition (example: Custom Value 1).
If customer Value is 1 the below logo will be displayed, instead of the original.
If it does not work try to remove <conditions> and keep only <condition>.
Note: You should not use event assignment ID as the logo would be changed only after you passed the first question.
	<settings>
	<css>
	<conditions>
	<condition>
	<customer>
	<customvalue1>1</customvalue1>
	</customer>
	</condition>
	</conditions>
	<value>
	.HeaderImage img { display: none; }
	.HeaderImage
	{
	background: url('https://mresearch.modernsurvey.com/images/2016551632341.Supervalu_Logo_Small.png') no-repeat;
	}
	</value>
	</css>
	</settings>

Items Features

Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.

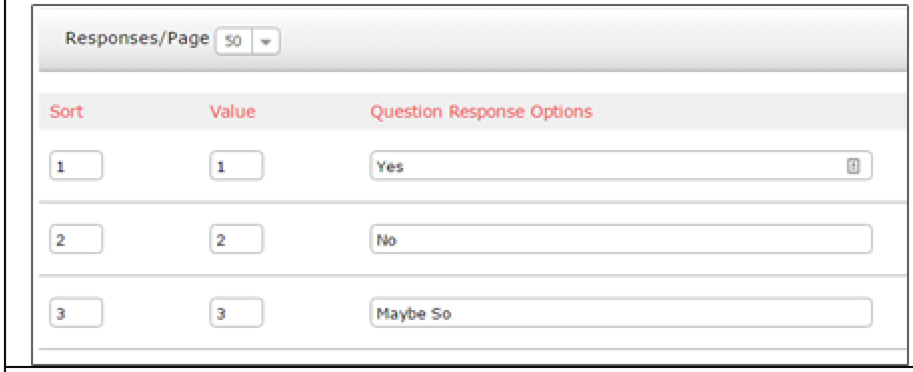
XML Example

The <item itemRef> is the beginning tag and should have a close tag </items>.

1. Text box validation against a list

Use Case: Question 1 on a survey is a passcode and it needs to be a number. Looks at the value of the text response.

1. Set up a drop down list in the survey designer of Research
   - The survey taker will see a textbox and must type in one of the values identified as a response option for the drop down item.
2. Identify the item as a type “4” in XML on the companion page.



Responses/Page 50

Sort	Value	Question Response Options
1	1	Yes
2	2	No
3	3	Maybe So

```
<items>
<item itemRef="1">
<itemType>4</itemType>
</item>
</items>
```

2. Series of text boxes for ranking

Use Case: Pick your top three items for leaving the company.

Create another row with this: Could also be used for checkbox (remove Item Type 5 row, Allow repeats and require each value statement) (You can check 1 to 3 options).

```
<items>
<item itemRef="2">
<settings>
<minValue>1</minValue> Range of values
<maxValue>3</maxValue> Range of values
</settings>
</item>
</items>
```

Items Features

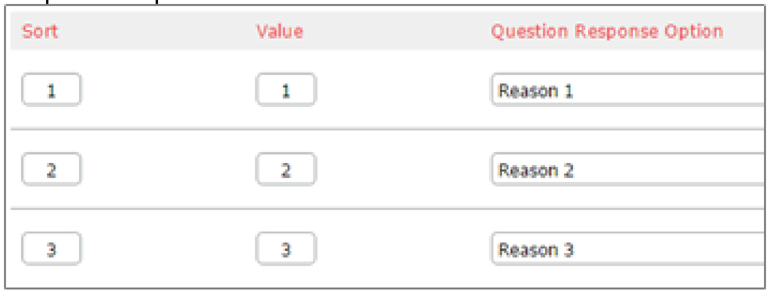
Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.

3. Series of text boxes for ranking with more options

Use Case: Pick your top three reasons for leaving the company.

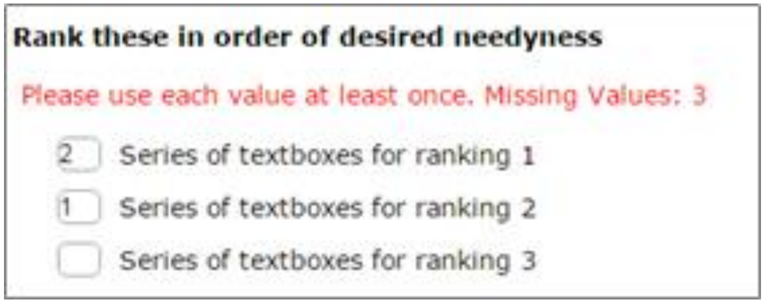
1. (Survey designer) Make a new radio item for each option to be ranked
2. (Survey designer) Create a scale and assign an item to each:
   Reason 1
   Reason 2
   Reason 3
3. (Survey designer) Add a section header for the item text
4. (Survey designer) Put a section break (or page break) after the last item to be ranked
5. (XML on the companion page) Identify the Q number of the first item
6. (XML on the companion page) Set the minValue and maxValue of the possible ranks – (1st, 2nd, 3rd in the above)
7. (XML on the companion page) “AllowRepeats” is either a ‘0’ for don’t allow or a ‘1’ for allow (if allowed you could have more than one item marked as 1st for instance)
8. (XML on the companion page) “requireEachValue” is either a ‘1’ for when you need each rank to be chosen at least once or a ‘0’ for when you don’t.

Step 2 example



Sort	Value	Question Response Option
1	1	Reason 1
2	2	Reason 2
3	3	Reason 3

Result:



Rank these in order of desired needyness

Please use each value at least once. Missing Values: 3

2 Series of textboxes for ranking 1

1 Series of textboxes for ranking 2

Series of textboxes for ranking 3

XML Example

The <item itemRef> is the beginning tag and should have a close tag </items>.

```
<items>
<item itemRef="2">
<itemType>5</itemType>
<settings>
<minValue>1</minValue>
<maxValue>3</maxValue>
<allowRepeats>0</allowRepeats>
<requireEachValue>0</requireEachValue>
</settings>
</item>
</items>
```

XML Detailed Explanation

1. For Item 2: <item itemRef="2">
2. Use text box ranking: <itemType>5</itemType>
3. Settings start tag: <settings>
4. Minimum range of value of 1: <minValue>1</minValue>
5. Maximum range of value of 3: <maxValue>3</maxValue>
6. Don’t let a user select this twice: <allowRepeats>0</allowRepeats>
7. Require each value to be entered at least once. (0=False/1=true):
   <requireEachValue>0</requireEachValue>

Items Features

Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.

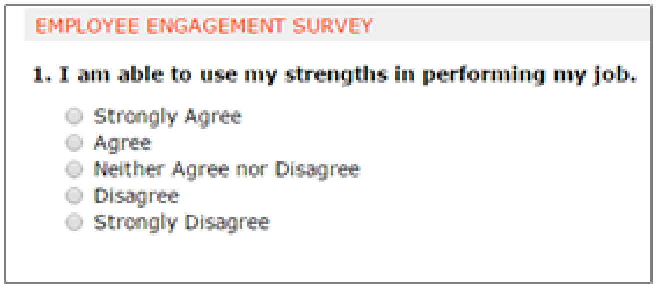
XML Example

The <item itemRef> is the beginning tag and should have a close tag </item>.

4. Blocking: Render as horizontal instead of vertical

Note: this does not work with Slider selectors.
In a survey using a ‘matrix’ style of organizing the items and responses (where there is a table with items down the left and response option across the top) sometimes single questions rendered in Research’s ‘vertical’ style can seem distracting. This can set any items to render in a ‘horizontal’ style like matrixed items.

Before:

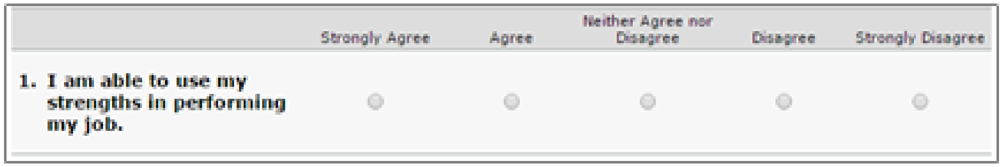


EMPLOYEE ENGAGEMENT SURVEY

1. I am able to use my strengths in performing my job.

○ Strongly Agree
○ Agree
○ Neither Agree nor Disagree
○ Disagree
○ Strongly Disagree

After:



	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1. I am able to use my strengths in performing my job.	○	○	○	○	○

```
<items>
<item itemRef="7">
<settings>
<renderAsBlock>1</renderAsBlock>
</settings>
</item>
```

5. Series of text boxes that must equal a certain amount

This item type can allow percentages to add up to 100% with validation.
Use Case Example: What percent of your time do you spend on each task?
Other summations are also possible.

1. (Survey designer) Create a text item for each response option
2. (Survey designer) Insert a section break after the last item
3. (XML on the companion page) Identify the “itemref” of the first item
4. (XML on the companion page) Make this item a type ‘6’
5. (XML on the companion page) Use “total” to identify what all the numbers are supposed to add up to
6. (XML on the companion page) Use “tolerance” to identify the leeway you want give (a ‘tolerance’ of 5 in the above example would let the taker’s answers add up to 95 or 105 instead of 100)

```
<items>
<item itemRef="8">
<itemType>6</itemType>
<settings>
<total>100</total>
<tolerance>0</tolerance>
</settings>
</item>
</items>
```

Items Features

Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.

6. The next questions are in random order

Use Case: When you don’t want the first question to always appear first.

Rules:

- Each item to be randomized is identified and has it’s randomizeType set to 1
- Each item must have the same scale

XML Example

The <item itemRef> is the beginning tag and should have a close tag </items>.

```
<items>
<item itemRef="18">
<settings>
<randomizeType>1</randomizeType>
</settings>
</item>

<item itemRef="19">
<settings>
<randomizeType>1</randomizeType>
</settings>
</item>

<item itemRef="20">
<settings>
<randomizeType>1</randomizeType>
</settings>
</item>
<items>
```

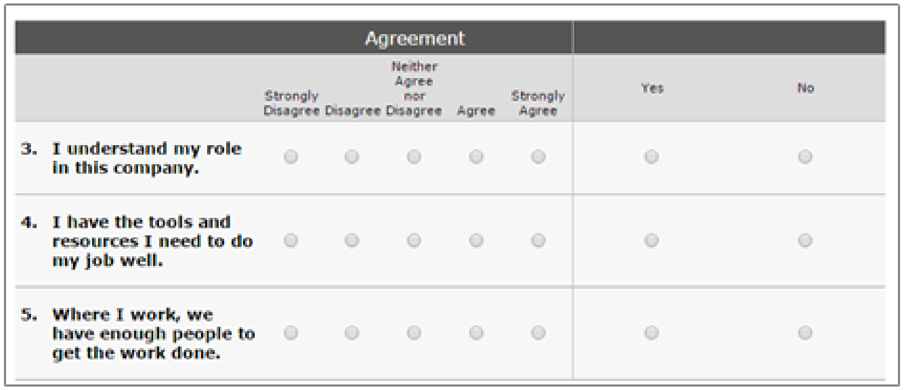
Items Features

Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.

7. Dual scale survey

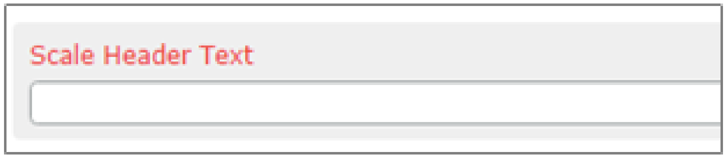
Use Case Example: Rate the current state and the ideal future state.
Note: Dual surveys do not support checkbox items or slider selectors.

1. (Survey designer) In the survey designer of Research create the items that you want to share in a dual scale matrix. (Example: If there should be 3 items in the dual scale, add 3 radio Question Types)
2. Enter the text for the items
3. Assign the desired scale for the for the first scale (Example: Agreement Scale)
4. 5Insert a Section Break
5. Click: Add More Questions
6. Add the same number of items as in step 1. (1 item for each question in the dual scale.)
7. Enter the same text for the items
8. Assign the desired scale for the second scale (Example: Yes/No)
9. (XML page) In the XML statement above change the first instance of itemRef to the Q number of the first item of the first scale and the second instance of itemRef to the Q number of the first item of the second scale
10. (XML Page) Make the itemType = 7



	Agreement						
	Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree	Yes	No
3. I understand my role in this company.	○	○	○	○	○	○	○
4. I have the tools and resources I need to do my job well.	○	○	○	○	○	○	○
5. Where I work, we have enough people to get the work done.	○	○	○	○	○	○	○

If you would like to update the Scale Header (Agreement), this can be done by modifying the Scale Header Text in the Scale Templates when you've created a custom scale.



Scale Header Text

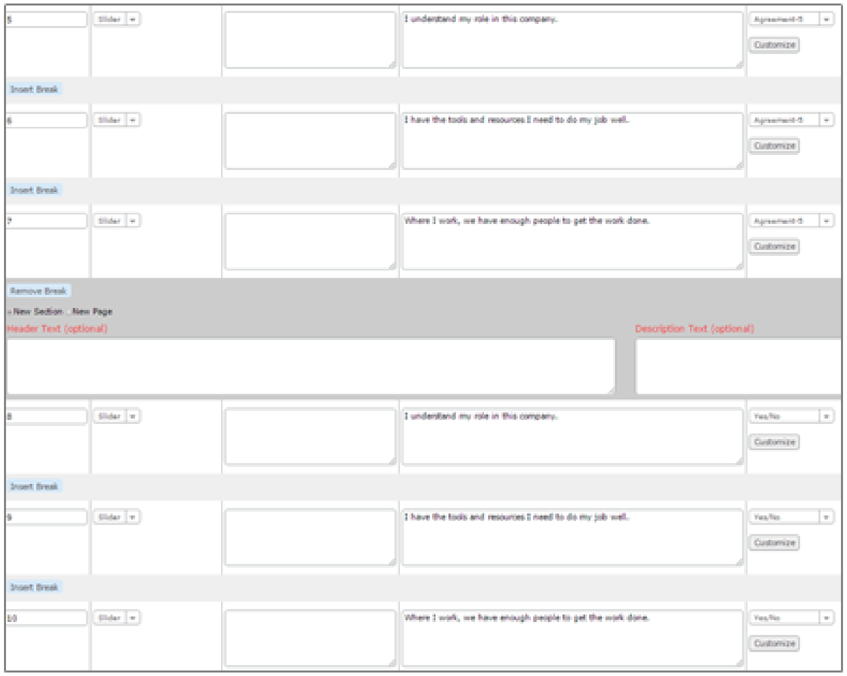
XML Example

The <item itemRef> is the beginning tag and should have a close tag </items>.

<items>
<item itemRef="5"> Starting at item 5, they are paired (5/8, 6/9, 7/10)
<itemType>7</itemType> Tells the system it the dual scale
<item itemRef="8"/>
</item>
</items>

<u>XML Detailed Explanation</u>

Starting at item 5, they are paired (5/8, 6/9, 7/10):<item itemRef="5">
Tells the system it the dual scale that item number 5 is paired to:
<itemType>7</itemType>



5 | Slider | I understand my role in this company. | Agreement-5 | Customize

Insert Break

6 | Slider | I have the tools and resources I need to do my job well. | Agreement-5 | Customize

Insert Break

7 | Slider | Where I work, we have enough people to get the work done. | Agreement-5 | Customize

Remove Break

○ New Section ○ New Page

Header Text (optional)

Description Text (optional)

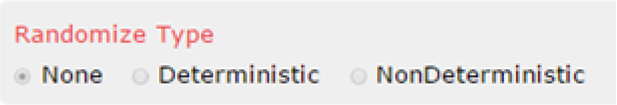
8 | Slider | I understand my role in this company. | Yes/No | Customize

Insert Break

9 | Slider | I have the tools and resources I need to do my job well. | Yes/No | Customize

Insert Break

10 | Slider | Where I work, we have enough people to get the work done. | Yes/No | Customize

Items Features Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.	**XML Example** The <item itemRef> is the beginning tag and should have a close tag </items>.
8. Randomize response options within a scale Response options can be randomized and certain options can be used identified as anchors that don't get randomized (like "other" or "I don't know") Anchoring (purple part in items tags) this in connection with randomizing entire survey allowsto anchor item 40 so it's not randomized. Typically anchoring is used to keep an open ended question at the end of the survey. Note: all items on the modify survey page that have unchecked "include on survey" need to be anchored in order to not impact survey display  Randomize Type ● None ○ Deterministic ○ NonDeterministic	<survey> <settings> <randomizeType>1</randomizeType> </settings> <items> <item itemRef="40"> <settings> <anchor>1</anchor> </settings> </item> </items> </survey>
9. Randomize the entire survey	<settings> <randomizeType>1</randomizeType> </settings>

Items Features Items Features allow you to manipulate how questions appear on the survey. These features will be between the items tags: <items></items>.	**XML Example** The <item itemRef> is the beginning tag and should have a close tag </items>.
10. Preselect a default value This will preselect a value and the surveyor can change it. Allows the user to quickly move through the survey. Use Case: Pre-determine the Opt-In/Opt-Out of individual engagement survey participation.	<items> <item itemRef="1"> <settings> <defaultQValue>3</defaultQValue> Radio slider dropdown </settings> </item> <item itemRef="2"> <settings> <defaultWValue>ABC</defaultWValue> Open or Text box </settings> </item> <item itemRef="2"> <settings> <defaultQMValue>1,2,4</defaultQMValue> Check box </settings> </item> </items>
11. Multi Select (Check) Use Case: Do not allow the user to pick conflicting selections. When you have 5 options and one is ‘None of the Above’ if the user clicks both None of the above and an option, it will clear all the options except the ‘none of the above’. Solo Option – Works with Multiselect.The 2nd items 5th value will be a Solo Option. If you select this all other will be deselected. You can also add multiple solo options within the same items. Example: Option 5 & 6	<items> <item itemRef="2"> <itemLabels> <itemLabel value="5" soloOption="true"/> </itemLabels> </item> </items> <items> <item itemRef="2"> <itemLabels> <itemLabel value="5" soloOption="true"/> <itemLabel value="6" soloOption="true"/> </itemLabels> </item> </items>

Filtering Features

Filtering features allow you to show or hide questions on the survey for different scenarios. These features will be between the filter tags: <filtering></filtering>.

1. Make an item disappear

Use Case: Hiding/Showing items based on a criteria such as the users organization.

1. Identify the value of the response options of the item that will drive the filtering
2. Use ‘itemRef’ to identify the Q number of the item that should disappear
3. Set the requirements that must be met for the filtering to occur
4. Like in branching the filtered item must be on a different page than the item or items identified as the requirement

XML Detailed Explanation
Example 1: (If Q22=2, hide Q23)

- For item 23: <item itemRef="23">
- If the condition is true: <condition>
- For the selection of this item:<item itemRef="22">
- If the user selects this response, hide item 23: <response value="2"/>

Example 2: (If Q22 is not 1 or 3, Hide Q24)
Item 22 & 24 must be on different pages for this to work.

- <item itemRef="24">
- <condition>
- If the user does not select 1 or 3, hide item 24: <item itemRef="22" inverse="true">
- <response value="1"/>
- <response value="3"/>

Example 3 using OR or And criteria (If Q22 =2 or Q18=1, hide Q23).

- Use And or OR: <item itemRef="23" operator="or">
- If the contition is True: <condition>
- For the selection of this item: <item itemRef="22">
- If the user selects this option hide item 23: <response value="2"/>
- </item>
- </condition>
- If the condition is true<condition>
- For the selection of this item: <item itemRef="18">
- If the user selects 1 hide item 23: <response value="1" />

XML Example

Example 1: (If Q22=2, hide Q23)
Item 22 & 23 must be on different pages for this to work

```
<item itemRef="23">
<condition>
<item itemRef="22">
<response value="2"/>
</item>
</condition>
</item>
```

Example 2: (If Q22 is not 1 or 3, Hide Q24)
Item 22 & 24 must be on different pages for this to work.

```
<item itemRef="24">
<condition>
<item itemRef="22" inverse="true">
<response value="1"/>
<response value="3"/>
</item>
</condition>
</item>
```

Example 3 using OR or And criteria (If Q22 =2 or Q18=1, hide Q23).

```
<item itemRef="23" operator="or">
<condition>
<item itemRef="22">
<response value="2"/>
</item>
</condition>
<condition> if the condition is true
<item itemRef="18">
<response value="1" />
</item>
</condition>
</item>
```

Filtering Features

Filtering features allow you to show or hide questions on the survey for different scenarios. These features will be between the filter tags: <filtering></filtering>.

XML Example

2. Filter an item when response has multiple options

Use Case: If item 8 is 1, hide response values 1, 2, 3 from item 10.

Use Min/Max for 1-3 response values

Do not do this:

```
<filtering>
  <itemLabel itemRef="10" >
      <value value = "1">
         <condition>
             <item itemRef="8">
                 <response value="1"/>
             </item>
          </condition>
       </value>
      <value value = "2">
         <condition>
            <item itemRef="8">
                <response value="1"/>
            </item>
         </condition>
       </value>
      <value value = "3">
         <condition>
            <item itemRef="8">
                <response value="1"/>
            </item>
         </condition>
       </value>
</itemLabel>
</filtering>
```

```
<filtering>
<itemLabel itemRef="10" >
<value min ="1"  max ="3">
<condition>
<item itemRef="8">
<response value="1"/>
</item>
</condition>
</value>
</itemLabel>
</filtering>
```

3. Filter an item based on custom values

Filtering hides response options and items, it does not jump pages.

```
<filtering>
<item itemRef="44">
<condition>
<customer>
<customvalue2 inverse="true">1</customvalue2> AND
<customvalue1>1</customvalue1>
</customer>
</condition>
</item>
</filtering>
```

Branching Features Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.	**XML Example**
Branching jumps from one question to another or to a different page. In cases where you want jump through all the remaining questions it is suggested to have text in the last item so the respondents have the opportunity to submit. AutoResponse automatically populates a question. May jump pages / items, but always use with branching if this is the desired outcome. Example: If there is no response, it will not jump the item, even if it jumps if the other criteria's are true.	
1. Advanced features Use Case: At a specific item, based on the response send the user to another page or question. Primarily used when you want to control when there's no answer and the user continues (the survey designer does not support this option). There must be a page break between the item ref and the destination. 1. Identify the item from which you want to branch 2. Set the item type to "basic" 3. Set what you want to happen for each value that can be chosen including; "none" for when no choice was made and "end" for when you want the taker to go to the end of the survey to submit the survey.	<branching> <item type="basic" itemRef="26"> For item 26 <response value="none" destinationItemRef="end" /> <response value="1" destinationItemRef="30" /> </item> </branching> **XML Detailed Explanation** For item 26: <item type="basic" itemRef="26"> If no response, send the user to the end of survey:<response value="none" destinationItemRef="end" /> If response value is 1, send the user to item 30:<response value="1" destinationItemRef="30" />
2. Branching to the finish & survey submission Use Case: Disqualifies a user from participation and submits the record so they cannot return and continue surveying. This code allows you to branch a respondent to the finish page by entering "finish' in the destination item ref. This is very similar to the previous branching to "end" but this feature submits the survey and doesn't allow the respondent to go back and change their answer.	<branching> <item itemRef="5" destinationItemRef="finish"> <condition> <item itemRef="5"> <response value="1"/> </item> </condition> </item> </branching> **XML Detailed Explanation** For item 5: <item itemRef="5" destinationItemRef="finish"> If the response is 1, branch to the finish page and submit: <item itemRef="5"> <response value="1"/>
3. Advanced branching	<branching> <item itemRef="5" destinationItemRef="10" >

Branching Features Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.	**XML Example**
If Q1=1 when you get to the page with item 5, then branch to Q10.	<condition> <item itemRef="1"> <response value="1" /> </item> </condition> </item> </branching>
4. Branching with no response Branch from item 20 to item 22 if somebody response 4-5-6 or does not respond.	<branching> <item itemRef=”20” destinationItemRef= “22” > <condition> <item itemRef=”6” operator=”or”> <response value=”none” /> <response value=”4” /> <response value=”5” /> <response value=”6” /> </item> </condition> </item> </branching>
5. Branching with no response or specific responses If item 1 has no response or a response of 3,4,5 go to item 10 Use Min/Max for 3-5 response values **Do not do this:** <survey> <branching> <item itemRef="1" type ="basic" > <response value="none" destinationItemRef= "10"/> <response value= "3" destinationItemRef= "10"/> <response value= "4" destinationItemRef= "10"/> <response value= "5" destinationItemRef= "10"/> </item> </branching> </survey>	<survey> <branching> <item itemRef="1" type ="basic" > <response value="none" destinationItemRef= "10"/> <response min="3" max="5" destinationItemRef= "10"/> </item> </branching> </survey>

Branching Features Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.	**XML Example**
6. Branching using event assignments Branch from item 20 to item 50 if somebody uses one of the two event assignments. Rules: • With an OR condition, OR condition should be always one level up.	<branching> <item itemRef=”20” destinationItemRef=”50” > <condition> <customer operator=”or”> <eventassignmentid>1FEB02F6-92B0-44C1-9F9A-B1884A58CB2</eventassignmentid> <eventassignmentid>1FEB02F6-92B0-44C1-9F9A-B1884A58CB2</eventassignmentid> </customer> </condition> </item> </branching>
7. Take users to a specific item regardless of response Use Case: Anyone who gets to item 2 will go to Q121 regardless of what they answered.	<branching> <item itemRef="2" destinationItemRef="121"> </item> </branching>

Branching Features

Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.

XML Example

8. Conditional sets

Useful when you have a condition that's used multiple times throughout the XML and use a label to identify it.

You can refer to sets of conditions anywhere that needs to do condition checking. They can even be considered together (within a given container) or referenced from other condition nodes (as in the example below).

In the Complex example, at **item 2** you will branch to 5 if **item 1** is 1 or 2 OR if **item 2** is 3.
Then at **item 5**, you will branch to the end if both **item 1** is 1 or 2 AND **item 2** is 3 AND **item 5** is 1.
You can mix and match both condition sets and inline condition nodes.

- Employment Qualifier tags are examples of a label that can be used.

Simpler Example with one condition

```
<conditionSets>
<EmploymentQualifier>
<condition>
<item itemRef="1">
<response value="1" />
<response value="2" />
</item>
</condition>
</EmploymentQualifier>
</conditionSets>
<branching>
<item itemRef="5" destinationItemRef="20">
<condition conditionSet="EmploymentQualifier" />
</item>
</branching>
```

Complex Example

```
<conditionSets>
<EmploymentQualifier>
<condition>
<item itemRef="1">
<response value="1" />
<response value="2" />
</item>
</condition>
</EmploymentQualifier>
<EmploymentQualifierB>
<condition>
<item itemRef="2">
<response value="3" />
</item>
</condition>
</EmploymentQualifierB>
<SuperEmploymentQualifier operator="or">
<condition conditionSet="EmploymentQualifier" />
<condition conditionSet="EmploymentQualifierB" />
</SuperEmploymentQualifier>
</conditionSets>
<branching>
<item itemRef="2" destinationItemRef="5" >
<condition conditionSet="SuperEmploymentQualifier" />
</item>
<item itemRef="5" destinationItemRef="end" operator=”and” >
<condition conditionSet="EmploymentQualifier" />
<condition conditionSet="EmploymentQualifierB" />
<condition>
<item itemRef="5">
<response value="1" />
</item>
</condition>
</item>
</branching>
```

Branching Features Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.	**XML Example**
9. Conditional redirection Use Case: Custom Finish Text commonly used for kiosk. The page goes to the finishing page which could be the login page for the next user to take the survey. This code allows you to conditionally redirect a user. In this example, if someone selected value=10 on item 6 they would be redirected to www.modernsurvey.com after they hit submit survey. Content=5 The number of seconds (5) to wait before sending the user to the URL URL=the url they are sent to.	<survey> <settings> <finishText> <conditions> <condition> <item itemRef="6"> <response value="10" /> </item> </condition> </conditions> <value> <meta http-equiv="refresh" content="5;url=https://www.modernsurvey.com"> </value> </finishText> </settings> </survey>
10. Conditional finish page text Use Case: Determine what the user see's based on if they qualify for an individual engagement report. Send the User to a Thank you page, then to their report.	<survey> <settings> <finishText> <conditions> <condition> <item itemRef="6"> <response value="10" /> </item> </condition> </conditions> <value> Thanks! We value your opinion! </value> </finishText> </settings> </survey>

Branching Features Branching features allow you to redirect the user to a specific page or question when a specific criteria is met. These examples will be between the branching tags: <branching></branching>.	**XML Example**
11. Survey finish time Use Case: Code checks if the respondent completed the survey in less than 120 seconds. If it’s true it returns a custom finish text and/or has the ability to redirect to a special location	<survey> <settings> <finishText> <conditions> <condition> <customer> <secondsonsurvey comparison="lessthan">120</secondsonsurvey> </customer> </condition> </conditions> <value> <h3>You finished too fast</h3> </value> </finishText> </settings> </survey>

Conditional Features Conditional Features allow you to expand on existing features and add specific criteria using execution statements. These are examples of different types of conditions that can be used in conjunction with other XML.	**XML Example**
1. Event Assignment ID Use case: when you want to use event assignment ID as part of your condition.	<condition> <customer> <eventassignmentid>1FEB02F6-92B0-44C1-9F9A0B1884A58CB2 </eventassignmentid> </customer> </condition>
2. Item Response (Can be Tieback) Use case: when you want to use a response value as part of your condition.	<condition> <item itemRef="6"> <response value="6" /> </item> </condition>
3. Culture Code (language) Use case: when you want to use culture code as part of your condition.	<condition> <customer> <culture>en-US</culture > </customer> </condition>
4. Customer Value Use case: when you want to use custom value 1 or 2 as part of your condition.	<condition> <customer> <customvalue1>4</customvalue1> </customer> </condition> <condition> <customer> <customvalue2>7</customvalue2> </customer> </condition>
5. Operator ‘OR’ Condition This should always appear one level up in the XLM statement. Use case: When you want to use Operator OR as part of your condition Operator = Or means the custom value is either 4 or 7	<condition> <customer operator="or"> <customvalue1>4</customvalue1> <customvalue1>7</customvalue1> </customer> </condition>
6. Operator ‘AND’ Condition Use case: When you want to use And as part of you condition. Operator = and means item 6 must = 6 AND item 7 must = 5	<condition operator ="and" > <item itemRef="6" > <response value="6" /> </item> <item itemRef="7" > <response value="5" /> </item> </condition>

<table>
<tr><th>Conditional Features

Conditional Features allow you to expand on existing features and add specific criteria using execution statements. These are examples of different types of conditions that can be used in conjunction with other XML.</th><th>XML Example</th></tr>
<tr><td><b>7. Operator ‘ANY’ Condition</b>

<b>Rules:</b>
1. Can only be used with multi select (checkbox)</td><td>&lt;item itemRef="36" destinationItemRef="37">
<condition>
<item itemRef="36" operator="any">
<response value="1"/>
</item>
</condition>
</item&gt;</td></tr>
<tr><td><b>8. Operator ‘EXACT’ Condition</b>

<b>Rules</b>:
1. Can only be used with multi select (checkbox)
2. The selections must be exact to make the condition True</td><td>&lt;item itemRef="36" destinationItemRef="44">
<condition>
<item itemRef="36" operator="exactly">
<response value="2"/>
</item>
</condition>
</item&gt;</td></tr>
<tr><td><b>9. Email Address Validation</b>

Use Case: Validate the email address has the specific criteria of an email address (@, .com, etc.) and return a message when incorrect.

How to use Regular expressions is not covered in this document. However this Regular expression is specific to emails.
Additional uses: phone numbers, zip codes, social security numbers, etc.</td><td>&lt;items>
<item itemRef="4">
<validators>
<validator type="RegularExpression" errorMessage="Incorrect email format">
<regularExpression>^([\w-\+\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$</regularExpression>
</validator>
</validators>
</item&gt;</td></tr>
</table>

Code Optimization

Below are some examples of the best way to simplify code for different features.

XML Example

1. Filtering response options

Use Case: When you have 5 response options, show the response option based on a condition.

Do not do this:

```
<filtering>
<itemLabel itemRef="2" >
<value value="1">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
<value value="2">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
<value value="3">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
<value value="4">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
<value value="5">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
</itemLabel>
```

Optimized Code:

```
<filtering>
<itemLabel itemRef="2" >
<value min="1" max="3">
<condition>
<item itemRef="1" inverse="True">
<response value="1"/>
</item>
</condition>
</value>
</itemLabel>
<itemLabel itemRef="2" >
<value min="4" max="5">
<condition>
<item itemRef="1" inverse="True">
<response value="2"/>
</item>
</condition>
</value>
</itemLabel>
</filtering>
```

Code Optimization

Below are some examples of the best way to simplify code for different features.

2. Branching

Combine the destination into one.

Do not do this:

```
<item type="basic" itemRef="3">
<response value="none" destinationItemRef="5"/>
<response value="1" destinationItemRef="5"/>
</item>
<item type="basic" itemRef="3">
<response value="none" destinationItemRef="5"/>
<response value="2" destinationItemRef="5"/>
</item>
<item type="basic" itemRef="3">
<response value="none" destinationItemRef="5"/>
<response value="3" destinationItemRef="5"/>
</item>
```

XML Example

Optimized Code :

```
<branching>
<item type="basic" itemRef="3">
<response value="none" destinationItemRef="5"/>
</item>
</branching>
```

Code Optimization

Below are some examples of the best way to simplify code for different features.

3. Auto response: Example 1

IMPORTANT!!! When using auto response XML, if not used correctly you could erase or overwrite legitimate survey responses!!!

Combine conditions when the target itemRef update is the same.

Do not do this:

```
<autoResponse>
<condition>
<item itemRef="8">
<response value="none" />
<response value="99" />
</item>
</condition>
<targetItem itemRef="9">NULL</targetItem>
</autoResponse>
<autoResponse>
<condition>
<item itemRef="8">
<response value="none" />
<response value="99" />
</item>
</condition>
<targetItem itemRef="10">NULL</targetItem>
</autoResponse>
<autoResponse>
<condition>
<item itemRef="8">
<response value="none" />
<response value="99" />
</item>
</condition>
<targetItem itemRef="9">NULL</targetItem>
</autoResponse>
<autoResponse>
<condition>
<item itemRef="8">
<response value="none" />
<response value="100" />
</item>
</condition>
<targetItem itemRef="10">NULL</targetItem>
</autoResponse>
<autoResponse>
<condition>
<item itemRef="8">
<response value="none" />
<response value="100" />
</item>
</condition>
```

XML Example

Optimized Code :

```
<autoResponse>
<condition><item itemRef="8">
<response value="none" />
<response value="99" />
<response value="100" />
</item>
</condition>
<targetItem itemRef="9-10">NULL</targetItem>
</autoResponse>
```

Code Optimization

Below are some examples of the best way to simplify code for different features.

XML Example

4. Auto response: Example 2

Combine Conditions: For each unique Target Item Ref create one condition when the Target Item rows are the same.

Do not do this:

```
<autoResponse>
<condition>
<item itemRef="8">
<responsevalue="none"/>
<responsevalue="99"/>
</item>
</condition>
<targetItem itemRef="10">NULL</targetItem>
</autoResponse>
<autoResponse>
<condition>
<item itemRef="8">
<responsevalue="none"/>
<responsevalue="100"/>
</item>
</condition>
<targetItem itemRef="10">NULL</targetItem>
</autoResponse>
```

Optimized Code :

```
<autoResponse>
<condition>
<item itemRef="8">
<responsevalue="none"/>
<responsevalue="99"/>
<responsevalue="100"/>
</item>
</condition>
<targetItem itemRef="10">NULL</targetItem>
</autoResponse>
```