

# Intermediate Designer 1: Special Qs, Validation, and C#

## Overview

- Special question types (GPS, barcode, picture)
- Data validation
- Basic C# syntax for creating validation and enablement conditions

## Special question types

## Special question types

- 3 types for now:
  - GPS
  - Barcode
  - Picture

# Special question types: GPS

Designer

Question Type  
Geo Location



Take GPS coordinate

Tap to record GPS



Take GPS coordinate

12.4880962, 41.8823389

Tap to record GPS

Interviewer

# Special question types: Barcode

Designer

Question Type  
QR Barcode

Barcode of the gift given to the household

Scan



Interviewer

Barcode of the gift given to the household

9781464801334

Scan

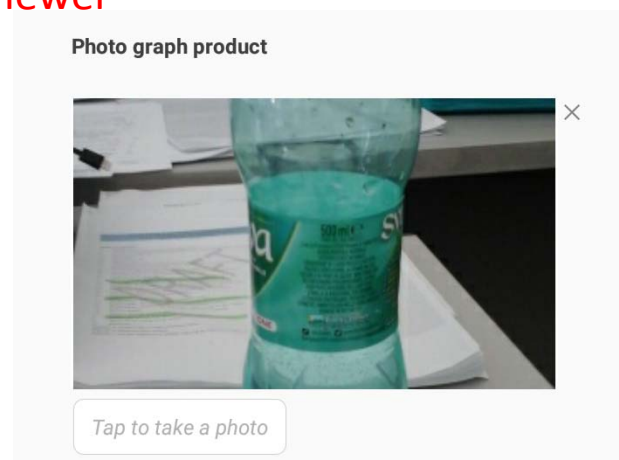
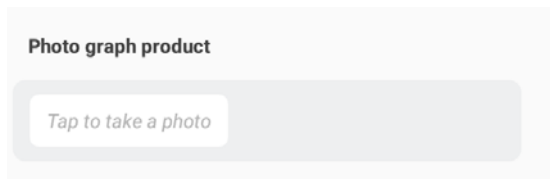
# Special question types: Photo

## Designer

Question type

 Picture 

## Interviewer



## Data validation

## Data validation: description

- Data validation expressions are to:
  - Confirm responses are consistent, and/or responses meet realistic expectations
  - Examples
    - A man cannot be pregnant. *\*Consistent w/ gender, enabled when household member is male.*
    - No one can be more than 110 years old, and younger than 0. *\*Realistic expectations.*
- These expressions are written in C# in the validation condition field.

## Data validation: description

- What happens when validation is violated?
- A warning message is shown to enumerator

# Data validation: description

MAIN RESPONDENT (ROSTER) / HOUSEHOLD MEMBER / BASICS /

Question type: 12 Numeric Variable name(?): s1\_age

Variable label(?): Age

Question text: How old is %rosteritle%? IN YEARS <br><font color="blue">Uneminyaka emingaki u%rosteritle%?</font>

Integer Roster upper bound: 110

ADD INTERVIEWER INSTRUCTION

Enabling condition (?):

Validation condition (?): `(s1_age < 110) && (s1_age > (-1))`

Error message (?): Age cannot be more than 110. Please check.

Mandatory question Question scope: Interviewer

SAVE CANCEL DELETE CLONE MOVE TO ▾

C#

Message enumerator sees if condition violated

# Data validation: description

HOLDING QUESTIONNAIRE / HOUSEHOLD MEMBER ROSTER: DEMOGRAPHIC CHARACTERISTICS AND EDUCATION /

Household member roster - Joe

Joe is

Male

Female

How old is Joe?

INSTRUCTION  
Please insert age in completed years

110

ANSWER IS INVALID  
Please check that age is inserted correctly

Value that violates condition

Message enumerator sees if condition violated

# Basic C# syntax for creating validation and enablement conditions

## Data validation: Building expressions

- How?
  - Connect a **variable name** with values or another variable using logical operators.
  - Connect expression creating multiple conditions using “||” or “&&”
- Examples
  - age > 18 || age < 65
  - gender == 2 && age > 14
  - \*Codes corresponding to options are used for single and multi-select questions**

# Data validation: Operators

Operator	Description
>	Greater than
<	Less than
==	Equal too
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to
&&	and
	or
<code>variableName==1    variableName==2</code>	(single select) contains in
<code>new decimal?[] {1,2}.Contains(variableName)</code>	(single select) contains in
<code>variableName.Contains(1)</code>	(multiselect) single values
<code>variableName.Intersect(new decimal[] {1, 3}).Any()</code>	(multiselect) single values, multi values

Operators depend on type of UNLEASH QUESTION!

## Building conditions

- Age greater than -1 and less than 100
  - Q: **M3\_Q6**
  - More than -1 and less than 100
  - Needed operators:
  - “>” “<” “&&”

Operator	Description
>	Greater than
<	Less than
==	Equal too
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to
&&	and
	or



# Data Validation examples

- Validation condition, we want the response to be less than 100 and positive.

● Validation condition 1 (?) ✕

`(M3_Q6 < 100) && (M3_Q6 > (-1))`

Error message (?)

Please verify the entered age of respondent is correct.

# Enablement and validation

Question type

12 Numeric ▼

Variable name(?)

⌘ M3\_Q22

Variable label(?)

Age of the manager

Question text

How old is %M3\_Q19%

Integer

- **Enable** only if there is a manager
- **Validate** that value is less one 100 and positive.

# Enablement and validation

UNLEASH

Question type:  Variable name(?):

Variable label(?):

Question text:

1	Yes	✕
2	No, the holder is the manager	✕

[ADD OPTION](#) [SHOW STRINGS](#)

RECIPIENT

Question type:  Variable name(?):

Variable label(?):

Question text:

Integer

# Enablement and validation

HOLDING QUESTIONNAIRE / HOLDING OVERVIEW /

Question type:  Variable name(?):

Variable label(?):

Question text:

Integer

Use 1000 separator (?)

Instruction (?)  Hide instruction (?)

✕

Enabling condition (?)  Hide if disabled (?)

Validation condition 1 (?) ✕

Error message (?)

# Enablement: Another Example

HOLDING QUESTIONNAIRE / LABOR OVERVIEW /

Question type Variable name(?)

**Categorical: Multi-select** M5\_Q9

Variable label(?)

Impact on farming activity

Question text

How did the labour shortage affect your farming activity?

1	Reduce productivity	×
2	Limit cultivated land	×
3	Reduce product quality	×
4	Reduce marketable surplus	×
5	Other	×

ADD OPTION SHOW STRINGS

## Data validation: Operators

Operator	Description
>	Greater than
<	Less than
==	Equal too
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to
&&	and
	or
<code>variableName==1    variableName==2</code>	(single select) contains in
<code>new decimal?[] {1,2}.Contains(variableName)</code>	(single select) contains in
<code>variableName.Contains(1)</code>	(multiselect) single values
<code>variableName.Intersect(new decimal[] {1, 3}).Any()</code>	(multiselect) single values, multi values

Operators depend on type of UNLEASH QUESTION!

# Enablement and validation

Question text

If other, please specify

Pattern (?)

[ADD INTERVIEWER INSTRUCTION](#)

Enabling condition (?)  Hide if disabled (?)

M5\_Q9.Contains(5)

[ADD NEW VALIDATION RULE](#)

# Questions?