Title

odkmeta (version 1.1.0) —— Create a do-file to import ODK data

Syntax

odkmeta using *filename*, **csv(***csvfile***) survey(***surveyfile*, *surveyopts***)**
choices(*choicesfile*, *choicesopts***)** [*options*]

options	Description
Main	
* **csv(***csvfile***)**	name of the .csv file that contains the ODK data
* **survey(***surveyfile*, *surveyopts***)**	import metadata from the *survey* worksheet *surveyfile*
* **choices(***choicesfile*, *choicesopts***)**	import metadata from the *choices* worksheet *choicesfile*
Fields	
dropattrib(*headers***)**	do not import field attributes with the column headers *headers*
keepattrib(*headers***)**	import only field attributes with the column headers *headers*
relax	ignore fields in *surveyfile* that do not exist in *csvfile*
Lists	
other(*other***)**	Stata value of **other** values of **select or_other** fields; default is **max**
oneline	write each list on a single line
Options	
replace	overwrite existing *filename*

* **csv()**, **survey()**, and **choices()** are required.

surveyopts	Description
Main	
type(*header***)**	column header of the *type* field attribute; default is **type**
name(*header***)**	column header of the *name* field attribute; default is **name**
label(*header***)**	column header of the *label* field attribute; default is **label**
disabled(*header***)**	column header of the *disabled* field attribute; default is **disabled**

choicesopts	Description
Main	
listname(*header***)**	column header of the *list_name* list attribute; default is **list_name**
name(*header***)**	column header of the *name* list attribute; default is **name**
label(*header***)**	column header of the *label* list attribute; default is **label**

other	Description
max	maximum value of each list: maximum list value plus one
min	minimum value of each list: minimum list value minus one
#	constant value for all value labels

Description

odkmeta creates a do-file to import ODK data, using the metadata from the *survey* and *choices* worksheets of the XLSForm. The do-file, saved to *filename*, completes the following tasks in order:

- Import lists as value labels
- Add **other** values to value labels
- Import field attributes as characteristics
- Split **select_multiple** variables
- Drop **note** variables
- Format **date**, **time**, and **datetime** variables
- Attach value labels
- Attach field labels as variable labels and notes
- Merge repeat groups

After **select_multiple** variables have been split, tasks can be removed from the do-file without affecting other tasks. User-written supplements to the do-file may make use of any field attributes, which are imported as characteristics.

Remarks

The **odkmeta** do-file uses **insheet** to import data. Fields that are long strings of digits, such as **simserial** fields, will be imported as numeric even if they are more than 16 digits. As a result, they will lose precision.

The do-file makes limited use of Mata to manage variable labels, value labels, and characteristics and to import field attributes and lists that contain difficult characters.

The do-file starts with the definitions of several local macros; these are constants that the do-file uses. For instance, local macro **`datemask'** is the mask of date values in the .csv files. The local macros are automatically set to default values, but they may need to be changed depending on the data.

Remarks for field names

ODK field names follow different conventions from Stata's constraints on variable names. Further, the field names in the .csv files are the fields' "long names," which are formed by concatenating the list of the *groups* in which the field is nested with the field's "short name." ODK long names are often much longer than the length limit on variable names, which is 32 characters.

These differences in convention lead to three kinds of problematic field names:

1. Long field names that involve an invalid combination of characters, for example, a name that begins with a colon followed by a number. **insheet** will not convert these to Stata names, instead naming each variable **v** concatenated with a positive integer, for example, **v1**.
2. Long field names that are unique ODK names but when converted to Stata names and truncated to 32 characters become duplicates. **insheet** will again convert these to **v**# names.
3. Long field names of the form **v**# that become duplicates with other variables that cannot be converted, for which **insheet** chooses **v**# names. These will be converted to different **v**# names.

Because of problem 3, it is recommended that you do not name fields as **v**#.

If a field name cannot be imported, its characteristic **Odk bad name** is **1**; otherwise it is **0**.

Most tasks that the **odkmeta** do-file completes do not depend on variable names. There are two exceptions:

1. The do-file uses **split** to split **select_multiple** variables. **split** will result in an error if a **select_multiple** variable has a long name or if splitting it would result in duplicate variable names.
2. The do-file uses **reshape** and **merge** to merge repeat groups. **reshape** will result in an error if there are long variable names. The merging code will result in an error if there are duplicate variable names in two datasets.

Where variable names result in an error, renaming is left to the user. The section of the do-file for splitting is preceded by a designated area for renaming. In the section for reshaping and merging, each repeat group has its own area for renaming.

Many forms do not require any variable renaming. For others, only a few variables need to be renamed; such renaming should go in the designated areas. However, some forms, usually because of many nested groups or groups with long names, have many long field names that become duplicate Stata names (problem 2 above). In this case, it may work best to use fields' short names where possible. The following code attempts to rename variables to their field short names. Place it as-is before the renaming for **split**:

```
foreach var of varlist _all {
    if "`:char `var'[Odk_group]'" != "" {
        local name = "`:char `var'[Odk_name]'" + ///
            cond(`:char `var'[Odk_is_other]', "_other", "") + ///
            "`:char `var'[Odk_geopoint]'"
        local newvar = strtoname("`name'")
        capture rename `var' `newvar'
    }
}
```

Remarks for lists

ODK list names are not necessarily valid Stata names. However, **odkmeta** uses list names as value label names, and it requires that all ODK list names be Stata names.

ODK lists are lists of associations of names and labels. There are two broad categories of lists: those whose names are all integer and those with at least one noninteger name. In the former case, the values of the value label are the same as the names of the list. In the latter, the values of the value label indicate the order of the names within the list: the first name will equal **1**, the second **2**, and so on. For such lists, the value of the value label may differ from the name of the list even if the name is a valid value label value; what matters is whether all names of the list are integer.

However, the value labels of these lists are easy to modify. Simply change the values of the value labels in the do-file; the rest of the do-file will be unaffected. Do not change the value label text.

Certain names do not interact well with **insheet**, which the **odkmeta** do-file uses to import the data.

For instance, it is not always possible to distinguish a name of **"."** from **sysmiss**. When it is unclear, the do-file assumes that values equal the name **"."** and not **sysmiss**. The problem arises when **insheet** imports **select** fields whose names in the data are the same as the values of a Stata numeric variable: real numbers, **sysmiss**, and extended missing values. **insheet** imports such fields as numeric, converting blank values (**""**) as **sysmiss**, thereby using the same Stata value for the name **"."** and for blank values.

insheet does not always interact well with list values' names that look like numbers with leading zeros, for example, **01** or **0.5**. If **insheet** imports a **select** field as numeric, it will remove such leading zeros, leading to incorrect values or an error in the do-file. For similar reasons, trailing zeros after a decimal point may be problematic.

List values' names that look like decimals may also not interact well with **insheet**. If **insheet** imports a **select** field as numeric, the do-file will convert it to string. However, for precision reasons, the resulting string may differ from the original name if the decimal has no exact finite-digit representation in binary.

Generally, names that look like numbers that cannot be stored precisely as **double** are problematic. This includes numbers large in magnitude.

Remarks for ODK variants

odkmeta is not designed for features specific to ODK variants, such as SurveyCTO or formhub. However, it is often possible to modify the **odkmeta** do-file to account for these features, especially as all field attributes are imported as characteristics.

SurveyCTO

For instance, the **odkmeta** do-file will result in an error for SurveyCTO forms that contain dynamic choice lists. One solution is to make the following changes to the do-file in order to import **select** fields with dynamic lists as string variables.

One section of the **odkmeta** do-file encodes **select** fields whose list contains a noninteger name. Here, remove dynamic lists from the list of such lists:

```
* Encode fields whose list contains a noninteger name.
local lists list1 list2 list3 ...
...
```

Above, if **list3** were a dynamic list, it should be removed.

The next section of the do-file attaches value labels to variables:

```
* Attach value labels.
ds, not(vallab)
if "`r(varlist)'" != "" ///
    ds `r(varlist)', has(char Odk_list_name)
foreach var in `r(varlist)' {
    if !`:char `var'[Odk_is_other]' {
...
```

Add a line to the second **if** command to exclude fields whose *appearance* attribute contains a **search()** expression:

```
* Attach value labels.
ds, not(vallab)
if "`r(varlist)'" != "" ///
    ds `r(varlist)', has(char Odk_list_name)
foreach var in `r(varlist)' {
    if !`:char `var'[Odk_is_other]' & ///
        !strmatch("`:char `var'[Odk_appearance]'", "*search(*)*") {
...
```

The do-file will now import fields with dynamic lists without resulting in an error.

formhub

formhub does not export **note** fields in the .csv files; specify option **relax** to **odkmeta**.

formhub exports blank values as **"n/a"**. Multiple sections of the **odkmeta** do-file must be modified to accommodate these.

Immediately before this line in the section for formatting **date**, **time**, and **datetime** variables:

```
if inlist("`type'", "date", "today") {
```

add the following line:

replace `var' = "" if `var' == "n/a"

Immediately before this line in the section for attaching value labels:

replace `var' = ".o" if `var' == "other"

add the following line:

replace `var' = "" if `var' == "n/a"

These lines replace **"n/a"** values with blank (**""**).

Remarks for "don't know," refusal, and other missing values

ODK lists may contain missing values, including "don't know" and refusal values. These will be imported as nonmissing in Stata. However, if the lists use largely consistent names or labels for the values, it may be possible to automate the conversion of the values to extended missing values in Stata. The following SSC programs may be helpful:

labmvs	**ssc install labutil2**
labmv	**ssc install labutil2**
labrecode	**ssc install labutil2**
labelmiss	**ssc install labelmiss**

Options

Main

survey(*surveyfile*, *surveyopts***)** imports the field metadata from the XLSForm's *survey* worksheet. **survey()** requires *surveyfile* to be a comma-separated text file. Strings with embedded commas, double quotes, or end-of-line characters must be enclosed in quotes, and embedded double quotes must be preceded by another double quote.

Each attribute in the *survey* worksheet has its own column and column header. Use the suboptions **type()**, **name()**, **label()**, and **disabled()** to specify alternative column headers for the *type*, *name*, *label*, and *disabled* attributes, respectively. All field attributes are imported as characteristics.

If the *survey* worksheet has duplicate column headers, only the first column for each column header is used.

The *type* characteristic is standardized as follows:

- **select one** is replaced as **select_one**.
- **select or other** is replaced as **select or_other**: **select_one** *list_name* **or other** is replaced as **select_one** *list_name* **or_other**, and **select_multiple** *list_name* **or other** is replaced as **select_multiple** *list_name* **or_other**.
- **begin_group** is replaced as **begin group**; **end_group** is replaced as **end group**; **begin_repeat** is replaced as **begin repeat**; and **end_repeat** is replaced as **end repeat**.

In addition to the attributes specified in the *survey* worksheet, **odkmeta** attaches these characteristics to variables:

Odk_bad_name is **1** if the variable's name differs from its ODK field name and **0** if not. See the remarks for field names above.

Odk_group contains a list of the *groups* in which the variable is nested, in order of the *group* level.

Odk_long_name contains the field's "long name," which is formed by concatenating the list of the *groups* in which the field is nested with the field "short name," with elements separated by **"-"**.

Odk_repeat contains the (long) name of the repeat group in which the variable is nested.

Odk_list_name contains the name of a **select** field's list.

Odk_or_other is **1** if the variable is a **select or_other** field and **0** if not.

Odk_is_other is **1** if the variable is a free-text **other** variable associated with a **select or_other** field; otherwise it is **0**.

For **geopoint** variables, **Odk_geopoint** is the variable's **geopoint** component: **Latitude**, **Longitude**, **Altitude**, or **Accuracy**. For variables that are not type **geopoint**, **Odk_geopoint** is blank.

choices(*choicesfile***,** *choicesopts***)** imports the list metadata from the XLSForm's *choices* worksheet. **choices()** requires *choicesfile* to be a comma-separated text file. Strings with embedded commas, double quotes, or end-of-line characters must be enclosed in quotes, and embedded double quotes must be preceded by another double quote.

Each attribute in the *choices* worksheet has its own column and column header. Use the suboptions **listname()**, **name()**, and **label()** to specify alternative column headers for the *list_name*, *name*, and *label* attributes, respectively. List attributes are imported as value labels.

If the *choices* worksheet has duplicate column headers, only the first column for each column header is used.

Fields

dropattrib(*headers***)** specifies the column headers of field attributes that should not be imported as characteristics. **_all** specifies that all characteristics be dropped.

keepattrib(*headers***)** specifies the column headers of field attributes to import as characteristics. **_all** means all column headers. Other attributes are not imported.

relax specifies that fields mentioned in *surveyfile* that do not exist in *csvfile* be ignored. By default, the do-file attempts to attach the characteristics to these variables, resulting in an error if the variable does not exist. For fields associated with multiple variables, for example, **geopoint** fields, **relax** attempts to attach the characteristics to as many variables as possible: an error does not result if some but not all variables exist.

Lists

other(*other***)** specifies the Stata value of **other** values of **select or_other** fields.

max, the default, specifies that the Stata value of **other** vary by the field's list. For each list, **other** will be the maximum value of the list plus one.

min specifies that the Stata value of **other** vary by the field's list. For each list, **other** will be the minimum value of the list minus one.

specifies a constant value for **other** that will be used for all lists.

oneline specifies that each list's value label definition be written on one line, rather than on multiple using **#delimit ;**.

Other

replace specifies that the **odkmeta** do-file be replaced if it already exists.

Examples

Create a do-file named **import.do** that imports ODK data, including the metadata in **survey.csv** and **choices.csv**

```
. odkmeta using import.do, csv("ODKexample.csv") survey("survey.csv")
    choices("choices.csv")
```

Same as the previous **odkmeta** command, but specifies that the field *name* attribute appears in the **fieldname** column of **survey_fieldname.csv**

```
. odkmeta using import.do, csv("ODKexample.csv")
    survey("survey_fieldname.csv", name(fieldname)) choices("choices.csv")
    replace
```

Same as the previous **odkmeta** command, but specifies that the list *name* attribute appears in the **valuename** column of **choices_valuename.csv**

```
. odkmeta using import.do, csv("ODKexample.csv")
    survey("survey_fieldname.csv", name(fieldname))
    choices("choices_valuename.csv", name(valuename)) replace
```

Create a do-file that imports all field attributes except for *hint*

```
. odkmeta using import.do, csv("ODKexample.csv") survey("survey.csv")
    choices("choices.csv") dropattrib(hint) replace
```

Same as the previous **odkmeta** command, but does not import any field attributes

```
. odkmeta using import.do, csv("ODKexample.csv") survey("survey.csv")
    choices("choices.csv") dropattrib(_all) replace
```

Create a do-file that imports **other** values of **select or_other** fields as **99**

```
. odkmeta using import.do, csv("ODKexample.csv") survey("survey.csv")
    choices("choices.csv") other(99) replace
```


Acknowledgements

Lindsey Shaughnessy of Innovations for Poverty Action assisted in almost all aspects of **odkmeta**'s development. She collaborated on the structure of the program, was a very helpful tester, and contributed information about ODK.

Author

Matthew White, Innovations for Poverty Action
mwhite@poverty-action.org