



## Help - Designing forms - Core concepts

Below, please find a series of topics on how to design your forms using SurveyCTO.

### Introduction

SurveyCTO helps you to collect data using electronic *forms*. These forms can be surveys, registration forms, inspection checklists, or anything else you might imagine. The definitions for these forms – the specifications that define how they should look and how they should function – are contained in spreadsheets that you can directly edit in Microsoft Excel or Google Sheets. Rather than editing form definitions directly, however, you can use SurveyCTO's online drag-and-drop form designer to more easily create and edit your forms.

If you learn best by doing, you can jump right in with our getting-started checklist or even go it alone by just starting a new form on the Design tab. To learn more first, read on...

#### [Online form design vs. editing spreadsheets directly.](#)

There are two ways to edit your form definitions in SurveyCTO: online using our drag-and-drop form designer, or directly using Microsoft Excel or Google Sheets. There are deep parallels between the two methods – so, for example, the language used to label things in the online designer matches the field and column names used in the spreadsheet definition – so that you can go back and forth, using whichever editing method is best for your current needs.

If you're new to SurveyCTO, you probably want to use the online designer since it's just an easier, friendlier format. But if you're on a plane, in a very remote area, or otherwise without Internet access, then directly editing a spreadsheet form definition is the way to go. Spreadsheets are also great for large-scale copying-and-pasting, collaborating with others (even in real-time if you're using Google Sheets!), tracking changes, or just saving form definitions for the historical record.

When you start a new form in the Design tab's *Your forms and datasets* section, "Edit online" is the easiest way to start editing your new form – but there are also "Advanced download options" that allow you to download the new form's definition to your local computer or Google Drive. And even if you select "Edit online", you can always click *Export* in the upper-right to export and begin editing the spreadsheet form definition. Likewise, for existing forms on your server, there is a simple "Edit" option to edit online, plus a "Download" option to download the form definition. There are also options to "Upload" new or updated form definitions, again using the spreadsheets directly.

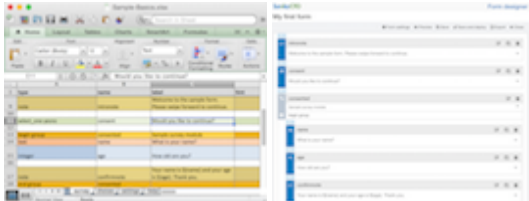
Much of the documentation here – in particular, the reference materials for all of the functions and field types available to you – is written from the perspective of the spreadsheet form definition. So, for example, each field type will be described in terms of the attributes that can be specified in the various columns of the form definition. All of the same attributes are available in the online designer, using largely the same language –

but it's all organized in a friendlier way. We have chosen not to separate this documentation for designer vs. spreadsheet users, in part because we expect that many users will eventually want to switch back and forth between the methods of form design. But, to help highlight information specific to those using the online form designer, some help content will appear in green.

Form definitions: an introduction

Your form's definition, in spreadsheet form, is a workbook with three worksheets:

- 1. The *survey* sheet, which specifies details about the data collected by your form (i.e., about the fields or questions in your form). In the online designer, the main window shows most fields and groups in this *survey* sheet (all except for standard meta-data fields, which are enabled or disabled in *Form settings*).



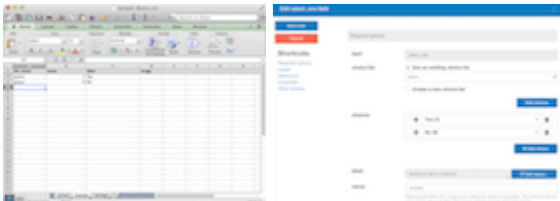
- 2. The *choices* sheet, which lists all of the option labels and internal values for multiple-choice questions. On this worksheet, the *list\_name* column specifies the name of a multiple-choice question type (without any spaces or punctuation, like "yesno"), the *value* column specifies the internal value stored for a given choice, and the *label* column specifies the label used for that choice in the form's user interface. In the online designer, a choice list is editable in-place, whenever you edit a multiple-choice question that uses it.

list_name	value	label	image
yesno	1	Yes	
yesno	0	No	

If you want to include an image for the choice label, you can specify an image filename (e.g., "yes.png") in the *image* column; if you do so, then you must always upload that image file as one of the attached media files when you upload the form to your SurveyCTO server. In the online designer, you simply attach images directly and they are automatically uploaded to the server, as part of your form definition.

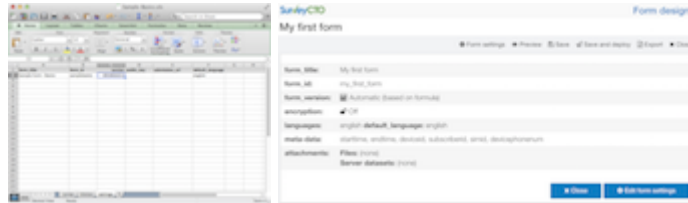
If you want to filter the list of choices shown to users based upon prior selections (e.g., to only show cities within the state already selected by the user), see *Dynamically filtering lists of multiple-choice options*.

See the *select\_one* and *select\_multiple* field types below for more on multiple-choice questions.



- 3. The *settings* sheet, which includes columns to hold the form title, form ID, and other form-wide settings. All settings should be recorded into row 2 of this worksheet, and the form ID must be unique

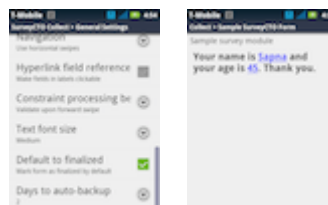
to the form (and not include any spaces or punctuation). Individual settings are documented elsewhere, with their relevant features. In the online designer, all form settings are available under *Form settings*.



You enter each question or field row-by-row into the *survey* sheet, in the sequence in which you want them to appear in your form. For each, the *type* column specifies the field or question type, and the *name* column specifies the internal name for that field. These field names must be unique, and they cannot include any spaces or punctuation. They are the variable names that will appear in your form's exported data.

For those field types that present a user interface, the *label* column is where you put the primary text for the question or field. Normally, your label is just plain text, but you can include HTML formatting if you wish to bold some text, use different colors, etc. – and the form designer includes a simple WYSIWYG HTML editor to make formatting easy. You can also enter line-breaks if needed; when using Excel to edit your form definition, you can insert line-breaks into a label by pressing Alt+Enter on Windows or Control+Option+Enter on a Mac (in the form designer, you just press Enter).

You can also refer to the data entered into or calculated by previous fields by including `${fieldname}` in your text, where "fieldname" is the name of the previous field. If you refer to a visible field (as opposed to a calculated one), then *SurveyCTO Collect* will hyperlink the field's value so that users can easily jump back to change it (if hyperlinking is enabled in *Collect*'s *General Settings*).



If you refer to a group of fields with `${groupname}`, then the label of the group will be displayed – and, if hyperlinking field references is enabled, that label will be hyperlinked to the start of the group. If the referenced group is a *repeat group*, then labels will be displayed for all instances in the repeat group. See the *Grouping and Repeating Questions* help topic below for more on grouping or repeating your fields.

The following topics cover everything else you need to know about designing forms in SurveyCTO. If you learn best by example, you can browse our library of sample forms.

## Starting a new form

To create a new form, navigate to the Design tab, scroll down to the *Your forms and datasets* section, and click + and then *Start new form* (or, if you want to duplicate an existing form, click the *Duplicate* option for that form). You will be able to start editing right away in the online form designer – or download a new form template to your local computer (as an Excel file) or to Google Drive (as a Google Spreadsheet). Either way, you will be asked for the following:

1. **Form title.** This is the visible title by which the form will be known to you and your team.

2. **Form ID.** This is the unique ID that will identify the form. While the form title can have spaces and can be changed from time to time, the form ID must not contain spaces and must remain fixed for the life of the form. The form ID must also begin with a letter, and can only include letters, numbers, underscores, and hyphens (no other punctuation or special characters).

### Other options

The new-form panel includes a few other options:

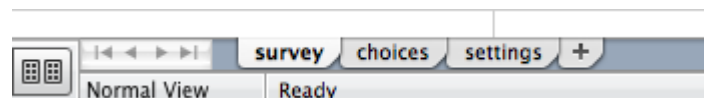
- *Use a sample form as your starting point?* If you enable this option, you will be able to select one of our sample forms as a starting point for your own form. It will be renamed with the form title and ID that you entered, but otherwise it will be a working sample, ready to customize.
- *Advanced option: Do you want this form's data to be encrypted?* If you check this box, then an additional step will allow you to attach a public encryption key to your form, so that all submitted data is safely encrypted. (See the *Encrypting form data* help topic for more on data encryption.)
- *Advanced option: Auto-generate fields necessary for pre-loading data?* If you check this box, then additional steps will guide you through some choices relevant to pre-loading data into your form; your form will then include the necessary fields to pre-load that data. (See the *Pre-loading data into a form* help topic for more on pre-loading data.)

### Getting started with your new form: using the designer

If you click the *Edit online* option at the end of the *Start new form* (or *Duplicate*) process, then you'll go straight into the online form designer. From there, you can easily add fields to your form, preview, save, and even deploy for data collection.

### Getting started with your new form: using the spreadsheet template

If you instead use one of the *Advanced download options* to download your new form as a template, then you will download an Excel workbook or Google Sheet with three worksheets in it, plus three quick-reference help worksheets.



One of those worksheets is a *settings* worksheet, which will have its cells pre-populated based on your form title, form ID, and encryption settings. The other two worksheets are where you will configure all of the specifics of your survey form. The *choices* worksheet specifies all valid choices for multiple-choice questions, and the *survey* worksheet specifies everything else about your survey form.

The default template includes a series of pre-configured fields at the top. These save details about when the survey was started and ended, how much time was spent in total, and the device on which the survey was filled out; none of these fields are ever seen by surveyors: they are saved automatically.

Your task is to add questions to this default template to make it your own. See the help topics below for details on the field types and techniques at your disposal. To get you started, there are a few example questions in the default template; you can start by either deleting or modifying them.

Once you have customized your form, go back to the *Your forms and datasets* section of the Design tab, click +, and then click *Upload form definition* to upload it to your server. Once the form is on your server, you can click *Preview* to preview it in your web browser, click *Fill out* on the Collect tab to actually submit data, or download the form onto a device by choosing *Get Blank Form* from the *SurveyCTO Collect* main menu. See the help topic on collecting data for more about the actual data-collection part of the process.

(By the way, you should maintain back-up copies of your form definitions for your records – but don't worry if you lose track of one. The *Your forms and datasets* section of the Design tab has a *Download* option you can use to download the form definition for any form on your server. You can even download a printable version of your form, if that's helpful.)

## Using the online form designer

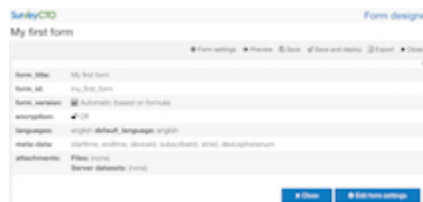
The SurveyCTO online form designer is integrated into the *Your forms and datasets* section of the Design tab. To use it, simply click *Edit* for any existing form, or click *Start new form* to start a new form and then click *Edit online* when you're ready to start editing it.

The main designer screen shows all of the fields and groups that exist in the form – except for a handful of invisible meta-data fields that are auto-filled by SurveyCTO (start time, end time, duration, username, etc.). Behind the scenes, these fields and groups are those that appear in the form definition's *survey* worksheet.



To view more information about a field or group, you can just click on its label (or on the little grey triangle to the right of its label). You can also click to edit, duplicate, or delete any field. And to re-order fields or move them into different groups, simply click, drag, and drop.

To edit your form's overall settings, click the *Form settings* button at the top of the designer window. This includes the form's title, unique ID, version, and other details typically recorded in a form definition's *settings* worksheet.



The designer's form-settings interface also allows you to do the following:

1. **Manage your form's languages.** This includes the name of the default language used in labels and images, as well as the names of any additional languages into which you'll be translating your form.
2. **Manage your form's attachments.** While you can easily add new file and dataset attachments as you add new media and/or pre-loaded data to your form, you can also go into form settings to add, update, or delete your form's attachments.
3. **Select which meta-data fields to include.** By default, new SurveyCTO forms include the following meta-data fields: starttime, endtime, deviceid, subscriberid, simid, devicephonenumber, username, duration, caseid. But you can choose which you want to include.

In the designer, you can always use, create, or edit the appropriate choice lists when adding or editing multiple-choice fields (in contrast to the spreadsheet form definition, where choice lists are off in the *choices* worksheet).



Once you have opened your form in the form designer, you can continue editing it even if your Internet connection has gone away. However, you will need to connect again before you can do any of the following: preview, save, save and deploy, or export (to Microsoft Excel or Google Sheets). To be safe, though, you should save as often as you can.

## Draft vs. deployed forms

You can preview and edit drafts of your survey forms from the Design tab, and only deploy them for actual data-collection when you're ready. Here's how it works:

- When you start a new form with *Start new form*, it always starts out as a draft. That means that nobody can see it, fill it out, or submit data for it yet. Until it's deployed, it will appear only in one place: in the *Your forms and datasets* section of the Design tab, highlighted in yellow. When you're ready to deploy, you can click *Deploy* there in the *Your forms and datasets* section, or click *Save and deploy* from the form designer. In the mean time, you can preview, test, and edit your form.
- Once you've deployed a form, you can edit it, save changes, preview and test those changes – all, again, without having to deploy your changes until you're ready. Until you instruct SurveyCTO to deploy something, it will remain in draft mode.
- And if you work with spreadsheet form definitions directly, you can always choose whether or not to immediately deploy whenever you upload a new or revised form definition.

## Organizing your forms

In the *Your forms and datasets* section of the Design tab, you can organize your forms in whatever way makes the most sense for you and your team.

Click + and then *Add group* to add groups or sub-groups, and then drag and drop forms, datasets, and groups in order to re-order or re-group them. (If you have a lot of re-organizing to do and dragging and dropping becomes too tedious, you can choose the *Organize* option at the top of the *Your forms and datasets* section; that will allow you to quickly move forms and datasets in bulk.)

Throughout the SurveyCTO server console, forms and datasets will appear grouped and ordered according to the Design tab. When *SurveyCTO Collect* and *SurveyCTO Sync* list all forms and/or datasets available on the server, they will also be listed in the same way; however, when *Collect* and *Sync* list forms that have been downloaded locally (for offline use), they will be shown in alphabetical order.

## Field types: text

This and the following help topics contain details on each of the field types that are available to you. You can click on the hyperlinks for pop-up examples of how these fields look in SurveyCTO Collect, on a little 3.1-inch Motorola MB632 screen.

In the form designer, click + *Add visible field*, then choose *text* as the field type.

**text:** prompts the user to enter text. Indicate "numbers", "numbers\_decimal", or "numbers\_phone" in the *appearance* column to limit entries to numeric input and prompt the numeric keypad to show; different devices show different numeric keypads with different capabilities (e.g., with the "numbers" appearance hyphens may not be allowed or may only be allowed in the first position), so you may need to try the three variations to see which allows the characters you want on your particular device. Use a text column with one of these "numbers" appearances if, for example, you want to accept a number greater than nine digits in length, or if you want to accept a hyphen-separated list of numbers.

In the spreadsheet form definition:

type	name	label	appearance
text	fieldname	question text	
text	fieldname	question text	numbers
text	fieldname	question text	numbers_decimal
text	fieldname	question text	numbers_phone

## Field types: integer

In the form designer, click + *Add visible field*, then choose *integer* as the field type.

**integer:** prompts the user to enter some whole number (no decimals). If you include "show\_formatted" in the *appearance* column, a formatted version of the number will be displayed as the user enters it (e.g., as in "120,000" for "120000"); any formatting will follow the Android device's locale settings to be appropriate to the local context.

Integers are limited, by nature, to be nine digits or less. If you need more than nine digits, use a *text* field with the "numbers" appearance (see above).

In the spreadsheet form definition:

type	name	label	appearance
integer	fieldname	question text	
integer	fieldname	question text	show_formatted

## Field types: decimal

In the form designer, click + *Add visible field*, then choose *decimal* as the field type.

**decimal:** prompts the user to enter some number (can include decimals). If you include "show\_formatted" in the *appearance* column, a formatted version of the number will be displayed as the user enters it (e.g., as in "120,000.43" for "120000.43"); any formatting will follow the Android device's locale settings to be appropriate to the local context.

In the spreadsheet form definition:

type	name	label	appearance
decimal	fieldname	question text	
decimal	fieldname	question text	show_formatted

## Field types: select\_one

In the form designer, click + *Add visible field*, then choose *select\_one* as the field type. You will be able to choose the choice list and edit choices as part of adding or editing the field.

**select\_one listname:** prompts the user to select one choice from a list of choices. When working with a spreadsheet form definition, "listname" must correspond to a value that appears in the *choices* worksheet's *list\_name* column (e.g., "yesno").

By default, choices will display as a series of radio buttons. Specify "quick" in the *appearance* column if you want to automatically advance to the next question as soon as an option is selected (without waiting for the user to swipe forward). Alternatively, specify the "minimal" appearance to show a single drop-down selector rather than the radio buttons, or the "compact" appearance to show a compact table of options. If you specify "compact", the number of columns will depend on the width of the display; to force a particular number of columns, specify the number of columns after a hyphen (as in "compact-3" for 3 columns). You can also combine the "quick" and "compact" appearances by specifying "quickcompact" or "quickcompact-#".

If you prefer to arrange your options horizontally, along a Likert-style scale, specify "likert" in the *appearance* column. If there isn't enough space to comfortably display all of the option labels, you can also specify "likert-min" to only show labels on either end, or specify "likert-mid" to only show labels at the left, right, and center (for an odd number of choices only). A note on display compatibility: when there are 10 or fewer options, each option will appear as a clickable radio button in web forms and on Android devices running Android 7.0 and above; on older Android devices and with more than 10 options, the individual radio buttons will not appear along the clickable scale.

If you would like to dynamically filter the list of choices shown to users, see the *Dynamically filtering lists of multiple-choice options* help topic. If you would like to dynamically load the list of choices from a dataset or .csv file, see the *Pre-loading data into a form* help topic.

In the spreadsheet form definition:

type	name	label	appearance
select_one listname	fieldname	question text	



select_one listname	fieldname	question text	likert
select_one listname	fieldname	question text	likert-min
select_one listname	fieldname	question text	likert-mid
select_one listname	fieldname	question text	quick
select_one listname	fieldname	question text	minimal
select_one listname	fieldname	question text	compact
select_one listname	fieldname	question text	compact-#
select_one listname	fieldname	question text	quickcompact
select_one listname	fieldname	question text	quickcompact-#

## Field types: select\_multiple

In the form designer, click + *Add visible field*, then choose *select\_multiple* as the field type. You will be able to choose the choice list and edit choices as part of adding or editing the field.

**select\_multiple listname:** prompts the user to select one *or more* choices from a list of choices. When working with a spreadsheet form definition, "listname" must correspond to a value that appears in the *choices* worksheet's *list\_name* column (e.g., "jobtypes").

By default, choices will display as a series of checkboxes. Specify "minimal" in the *appearance* column to use a single pop-up selector rather than a series of checkboxes, or specify "compact" to show a compact table of options. If you specify "compact", the number of columns will depend on the width of the display; to force a particular number of columns, specify the number of columns after a hyphen (as in "compact-3" for 3 columns).

If you would like to dynamically filter the list of choices shown to users, see the *Dynamically filtering lists of multiple-choice options* help topic. If you would like to dynamically load the list of choices from a dataset or .csv file, see the *Loading multiple-choice options from pre-loaded data* help topic.

When you want to use a *select\_multiple* field in field validation or skip patterns (i.e., in constraint or relevance expressions), you will need to use the *selected()* function to query whether a given choice has been selected. (See the help topics below to learn more about constraint and relevance expressions.)

When you export your data with *SurveyCTO Sync*, each row will include one column – containing a space-separated list of all chosen response values – for each *select\_multiple* field. If you would like each field to be exported also as a series of 1/0 columns for each possible answer value (i.e., as a series of "dummy variables"), go into *SurveyCTO Sync* and enable *Export select\_multiple responses as a series of 1/0 columns* in the preferences.

In the spreadsheet form definition:

type	name	label	appearance
------	------	-------	------------

select_multiple listname	fieldname	question text	
select_multiple listname	fieldname	question text	minimal
select_multiple listname	fieldname	question text	compact
select_multiple listname	fieldname	question text	compact-#

## Field types: geopoint

In the form designer, click + *Add visible field*, choose *geo...* as the field type, and leave the type set to *geopoint* when editing the field details.

**geopoint:** collects GPS coordinates using the device's built-in GPS support (or the web browser's location service). Indicate "maps" in the *appearance* column to immediately show the captured location on a map or "placement-map" in the *appearance* column to show the location and also allow the user to manually adjust that location using a pin (both map options generally require an Internet connection, but see this help topic for an advanced option to load offline map tiles).

Alternatively, specify "background" in the *appearance* column to invisibly record the user's location in the background. This option only works on Android since web browsers don't allow secret location detection – and even on Android you might not want to record locations invisibly since field users might need to go outside in order to capture accurate GPS readings.

On Android devices, *SurveyCTO Collect* will keep trying to acquire a more accurate GPS reading until it reaches a specific accuracy threshold (by default: +/- 5m accuracy). If you use the "background" appearance, Collect will keep trying to read the location until it reaches that target accuracy or the user exits the form, whichever comes first; if not using "background", users can manually stop the process and accept the current level of accuracy at any time. If you would like to use a different accuracy threshold than 5m, just specify the number of meters in the *accuracy\_threshold* column (for example, "1" to keep trying until 1m accuracy is reached, or "10" for 10m accuracy; only applies when form filled on Android devices).

In the spreadsheet form definition:

type	name	label	appearance	accuracy_threshold
geopoint	fieldname	question text		
geopoint	fieldname	question text	maps	
geopoint	fieldname	question text	placement-map	
geopoint	fieldname	question text	background	
geopoint	fieldname	question text	background	#

For more information on other GPS data collection options, GPS performance tips, or advanced options for offline map support, see *Collecting GPS data*. (As long as your device has a GPS sensor, you can generally collect *geopoint* data offline. It's just the "maps" and "placement-map" *appearance* options that require

maps – and therefore special considerations for offline use.)

#### Default hint

On Android devices, *geopoint* fields have a default hint that appears in italics below the label you specify: "GPS coordinates can only be collected when outside." If you want to specify a different hint (perhaps in a different language), you can specify your own text for the field's *hint*.

### Field types: geoshape

In the form designer, click + *Add visible field*, choose *geo...* as the field type, and then choose *geoshape* as the *type* when editing the field details.

**geoshape:** collects GPS coordinates that form a polygon enclosing some area, using maps and the device's built-in GPS support. You can easily view the collected polygons in SurveyCTO's *Data Explorer*, or export them, with the rest of your data, for use in outside GIS systems.

Identifying shapes on maps generally requires an Internet connection, but see this help topic for an advanced option to load offline map tiles.

In the spreadsheet form definition:

type	name	label
geoshape	fieldname	question text

#### Android usage instructions

Upon reaching a *geoshape* form field, begin the collection process by pressing the "Start GeoShape" button. This will load a world map and prompt you to choose where in the world to zoom ("Current location" is usually the best choice, unless editing data collected earlier). Once you have zoomed to the correct location on the map, long-press on the map to drop a pin and begin creating the shape. Long-press again at another location on the map to drop another pin and connect the points so far. Then, long-press to drop additional points along the desired boundary.

Each dropped pin will connect to the last one, and the last pin will also connect to the first, in order to close the polygon. If you make a mistake, you can long-press on an existing pin to pick it up and drag it to a new location. Click the trash-can button to clear and start over, or the disk icon to save and exit. (Screenshots here...)

#### Web usage instructions

Web forms do not currently support *geoshape* fields; if you preview or fill out a form on the web, these fields will fall back to just collecting a single GPS position. If your project requires better web forms support, please let us know by going to our online support center and either opening a request or posting a message in one of the community forums.

#### Default hint

On Android devices, *geoshape* fields have a default hint that appears in italics below the label you specify: "GPS coordinates can only be collected when outside." If you want to specify a different hint (perhaps in a different language), you can specify your own text for the field's *hint*.

## Field types: geotrace

In the form designer, click + *Add visible field*, choose *geo...* as the field type, and then choose *geotrace* as the *type* when editing the field details.

**geotrace**: collects GPS coordinates that form either an open polyline or a closed polygon enclosing some area, using maps and the device's built-in GPS support. Generally, you collect *geotrace* data by collecting GPS points as you walk or drive along some route or boundary. You can easily view the collected polylines or polygons in SurveyCTO's *Data Explorer*, or export them, with the rest of your data, for use in outside GIS systems.

When collecting *geotrace* data, you can choose either "manual" or "automatic" modes. In manual mode, you click a button at the top of the screen to drop a pin whenever you want. In automatic mode, a pin automatically drops every x seconds; you choose the x when you start, and you can still click the button to manually drop extra pins as you trace in automatic mode. When you save, you choose to save as a polyline (in which case the dropped pins are recorded, in order, as they are) or as a polygon (in which case the first and last points are automatically connected, to close a polygon).

Collecting data on maps generally requires an Internet connection, but see this help topic for an advanced option to load offline map tiles.

In the spreadsheet form definition:

type	name	label
geotrace	fieldname	question text

### Android usage instructions

Upon reaching a *geotrace* form field, begin the collection process by pressing the "Start GeoTrace" button. Then choose either "manual" or "automatic" modes – and, if you select automatic, choose how often to automatically drop pins, as you trace your route or boundary. Next, choose where in the world to zoom ("Current location" is usually the best choice, unless editing data collected earlier). Once you have zoomed to the correct location on the map, click the "Record GeoPoint" button at the top of the screen to drop a pin whenever you wish. If you had selected automatic operation, pins will drop automatically, every x seconds or minutes (according to your choice when you began); if needed, you can click the pause button on the right to pause the automatic pins.

Each dropped pin will connect to the last one. Click the trash-can button to clear and start over (pausing first, if in automatic mode), or the disk icon to save and exit. When you save, choose to save as a polyline (in which case the dropped pins are recorded, in order, as they are) or as a polygon (in which case the first and last points are automatically connected, to close a polygon). (Screenshots here...)

### Web usage instructions

Web forms do not currently support *geotrace* fields; if you preview or fill out a form on the web, these fields will fall back to just collecting a single GPS position. If your project requires better web forms support, please let us know by going to our online support center and either opening a request or posting a message in one of the community forums.

### Default hint

On Android devices, *geotrace* fields have a default hint that appears in italics below the label you specify: "GPS coordinates can only be collected when outside." If you want to specify a different hint (perhaps in a different language), you can specify your own text for the field's *hint*.

## Field types: barcode

In the form designer, click + *Add visible field*, then choose *barcode* as the field type.

**barcode**: scans a barcode. To use this field type, you must install the ZXing barcode scanner app onto each of your Android devices (it's not supported by the web forms interface, which will just display a *barcode* field as a simple text box). For generating barcodes in bulk, you might consider using the OnMerge Barcode product to generate barcodes in Microsoft Word.

In the spreadsheet form definition:

type	name	label
barcode	fieldname	question text

## Field types: datetime, date, and time

In the form designer, click + *Add visible field*, then choose *datetime* as the field type.

**datetime**: prompts the user to enter a date and time. For devices with smaller screens, specify "no-calendar" in the *appearance* column so that the user interface will be smaller. For the date part, specify "month" or "month-year" in the *appearance* column to only ask for the month or month+year.

In the spreadsheet form definition:

type	name	label	appearance
datetime	fieldname	question text	
datetime	fieldname	question text	no-calendar
datetime	fieldname	question text	month
datetime	fieldname	question text	month-year

Alternatively, in the form designer, click + *Add visible field*, then choose *date* as the field type.

**date**: prompts the user to enter a date. For devices with smaller screens, specify "no-calendar" in the *appearance* column so that the user interface will be smaller. To only ask for the month or month+year, specify "month" or "month-year" in the *appearance* column.

In the spreadsheet form definition:

type	name	label	appearance
------	------	-------	------------

date	fieldname	question text	
date	fieldname	question text	no-calendar
date	fieldname	question text	month
date	fieldname	question text	month-year

Alternatively, in the form designer, click + *Add visible field*, then choose *time* as the field type.

**time:** prompts the user to enter a time.

In the spreadsheet form definition:

type	name	label	appearance
time	fieldname	question text	

## Field types: image

In the form designer, click + *Add visible field*, then choose *image* as the field type.

**image:** prompts the user for an image. If you specify nothing in the *appearance* column, the user will be prompted to take a picture or to select an existing picture from the device.

*Important:* if users will be taking pictures, be sure to adjust the device settings so that the default picture resolution is not so high that picture files are extremely large. The larger the picture files, the longer it will take to upload form data to the SurveyCTO server (since the form data will include those picture files). Large photos can seriously slow and disrupt your data pipeline.

Specify "new" in the *appearance* column to require that the user take a new photo (vs. select an existing one). To allow users to annotate their taken or selected pictures, specify "annotate" in the *appearance* column. To allow users to submit free-form drawings or signatures, instead specify "draw" or "signature" in the *appearance* column. Note, however, that the "annotate", "draw", and "signature" appearances have no effect on web forms: those appearances only change the user experience when using the Android app.

In the spreadsheet form definition:

type	name	label	appearance
image	fieldname	question text	
image	fieldname	question text	new
image	fieldname	question text	annotate
image	fieldname	question text	draw
image	fieldname	question text	signature

## Field types: audio

In the form designer, click + *Add visible field*, then choose *audio* as the field type.

**audio:** prompts the user to select or record an audio clip. Audio clips may be large and therefore may take a longer time or require a high-speed connection to to upload form data to the SurveyCTO server. Specify "new" in the *appearance* column to require that the user record a new audio clip (vs. select an existing one).

In the spreadsheet form definition:

type	name	label	appearance
audio	fieldname	question text	
audio	fieldname	question text	new

Please note: different devices use different audio-recording software, and different software records in different formats. Audio recordings might save and export in a common format like MP4, or they might use a format more specific to Android, like 3GPP. If you have trouble opening and playing your audio files, you might need to install software like VLC that plays the format in which it was saved.

## Field types: video

In the form designer, click + *Add visible field*, then choose *video* as the field type.

**video:** prompts the user to record a video clip. Video clips may be large, so uploading form data with video clips may take a long time or require a high-speed connection. Specify "new" in the *appearance* column to require that the user record a new video clip (vs. select an existing one).

In the spreadsheet form definition:

type	name	label	appearance
video	fieldname	question text	
video	fieldname	question text	new

## Field types: file

In the form designer, click + *Add visible field*, then choose *file* as the field type.

**file:** prompts the user to attach a file. By default, users can attach only these file types: text, image, video, audio, PDF, ZIP, or MS Office files (Excel or Word). Alternatively, you can specify your own comma-separated list of MIME types for those types of files that you would like to accept; for example, this list includes the

MIME types for MS Word and Excel files: "application/vnd.ms-excel, application/msword, application/vnd.openxmlformats-officedocument.wordprocessingml.document, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet".

In the spreadsheet form definition:

type	name	label	mediatype
file	fieldname	question text	
file	fieldname	question text	application/vnd.ms-excel, application/msword, application/vnd.openxmlformats-officedocument.wordprocessingml.document, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

## Field types: note

In the form designer, click + *Add visible field*, then choose *note* as the field type.

**note:** displays text to the user. Note fields do not actually collect any data. They can be used, for example, to confirm entries that have already been made, or to provide additional interviewer instructions. The full text for the note goes into the *label* column.

If you include a note field with "intro" in its *appearance* column, that note will appear in the survey form's opening screen, before the first visible form field. If you include a note field with "thankyou" in its *appearance* column, that note will appear at the very end of your form, after data has been finalized and saved.

(Otherwise, the note field will appear in its proper position relative to other fields, based on where you place it in your form definition.)

In the spreadsheet form definition:

type	name	label	appearance
note	fieldname	note text	
note	fieldname	note text	intro
note	fieldname	note text	thankyou

## Field types: start

In the form designer, edit *Form settings*, then check the *starttime* meta-data field (if it isn't already).

**start:** record the date and time the survey was started. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

In the spreadsheet form definition:



type	name
start	fieldname

### Field types: end

In the form designer, edit *Form settings*, then check the *endtime* meta-data field (if it isn't already).

**end:** record the date and time the survey was ended. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

In the spreadsheet form definition:

type	name
end	fieldname

### Field types: deviceid

In the form designer, edit *Form settings*, then check the *deviceid* meta-data field (if it isn't already).

**deviceid:** record the unique ID of the device used to fill out the survey. This field type can be used to link different surveys filled out on the same device. For example, you might have a "login form" that surveyors fill out each morning when they begin using a device, and then you might use the *deviceid* to link that data with the data from later forms filled out on the same device. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

When users fill in forms using a web browser, the recorded *deviceid* will always be "(web)".

In the spreadsheet form definition:

type	name
deviceid	fieldname

### Field types: subscriberid

In the form designer, edit *Form settings*, then check the *subscriberid* meta-data field (if it isn't already).

**subscriberid:** record the subscriber ID associated with the device's SIM card, if any. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface. When users fill in forms using a web browser, this field will always be left blank.

In the spreadsheet form definition:

type	name
subscriberid	fieldname

### Field types: simserial

In the form designer, edit *Form settings*, then check the *simid* meta-data field (if it isn't already).

**simserial:** record the serial number associated with the device's SIM card, if any. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface. When users fill in forms using a web browser, this field will always be left blank.

In the spreadsheet form definition:

type	name
simserial	fieldname

### Field types: phonenumber

In the form designer, edit *Form settings*, then check the *devicephonenumber* meta-data field (if it isn't already).

**phonenumber:** record the phone number associated with the device's SIM card, if any. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface. When users fill in forms using a web browser, this field will always be left blank.

In the spreadsheet form definition:

type	name
phonenumber	fieldname

### Field types: username

In the form designer, edit *Form settings*, then check the *username* meta-data field (if it isn't already).

**username:** record the username of the user filling out the form. In *SurveyCTO Collect* on Android devices, this will be the username currently configured in Collect's settings (with a note if different from the identity last used to authenticate to the server); when users fill in forms with a web browser, it will either be the username they logged in with, or "(anonymous)" if they didn't log in at all. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

In the spreadsheet form definition:

type	name
------	------

username	fieldname
----------	-----------

## Field types: caseid

In the form designer, edit *Form settings*, then check the *caseid* meta-data field (if it isn't already).

**caseid:** record the unique ID of the case for which the form was filled out. This field will be blank unless the user filled out the form using the *Manage Cases* button or clicked a hyperlink that included a *caseid* parameter. See the help topic on case management for more on cases. This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

In the spreadsheet form definition:

type	name
caseid	fieldname

## Field types: comments

In the form designer, click + *Add hidden field*, then choose *comments* as the field type.

**comments:** allow users to enter free-form comments associated with any field(s) when filling out the form. When a field of this type is present in the form, users will have a new *Add comment* option available when filling out the form (depending on the version of Android, they can press the device's menu button and then *Add comment*, or just press the pencil button; in the web interface, it is an option on the *Options* menu at the top of the screen).

When you export data that includes comments, those comments will be exported into individual, submission-specific .csv files in the *media* subdirectory; the full path and filename for each .csv file will be included in the exported form data, as the data for your comments field. For example, if you have a comments field named *comments*, then the *comments* column in your exported data will, for each submission that includes comments, include a path and filename for a separate .csv file containing those comments; in that .csv file, there will be two columns, one containing the field name and one containing the comments on that field.

In the spreadsheet form definition:

type	name
comments	fieldname

## Field types: calculate and calculate\_here

In the form designer, click + *Add hidden field*, then choose *calculate* as the field type. You will be able to use a wizard to construct common calculation expressions, or edit the expression by hand for more complex or expert use.

**calculate:** calculate using prior field values. The value of this field will be calculated using the expression in the *calculation* column (e.g., "({age1}+{age2}+{age3}) div 3" to calculate an average age from three prior age fields). This is an automatic, hidden field type: it is filled in automatically and never shows up in the survey's user interface.

The easiest way to build a calculation expression for spreadsheet form definitions is to use the calculation-builder on the server's Design tab: for any new or existing form, select *Tools*, then *Build calculation*. To use the calculation-builder offline, simply run *SurveyCTO Sync* and choose the calculation-builder from the *Tools* menu's *Form tools* sub-menu.

For hand-creating or hand-editing calculation expressions, see the *Using expressions in your forms* section below.

In the spreadsheet form definition:

type	name	calculation
calculate	fieldname	expression

#### Advanced fixed-location calculations

In the form designer, click + *Add hidden field*, then choose *calculate\_here* as the field type. You will be able to use a wizard to construct common calculation expressions, or edit the expression by hand for more complex or expert use.

**calculate\_here:** same as *calculate* (immediately above), but *calculate\_here* calculations only happen when users reach the field's place in the survey. (In contrast, regular *calculate* fields calculate whenever forms load or save, and whenever fields upon which they rely change.)

**Please note that the two examples below may be the only legitimate uses of *calculate\_here* fields. The vast majority of your calculations should be in regular *calculate* fields.** This is because regular *calculate* fields automatically re-calculate when needed, which is almost always a good thing.

For one example where *calculate\_here* is useful, consider the case where you want to capture the number of seconds already spent filling or editing the form at the precise moment that the user first reaches a particular point: for this, you can use a *calculate\_here* field at that point, with "once(duration())" in its *calculation* column. Or, to capture the date and time at which the user first reaches a particular point in your form, use a *calculate\_here* field at that point, with "once(format-date-time(now(), '%Y-%b-%e %H:%M:%S'))" in its *calculation* column. For more on these and other expressions, see the *Using expressions in your forms* section below.

In the spreadsheet form definition:

type	name	calculation
calculate_here	fieldname	expression

## Field types: speed violations

In the form designer, click + *Add hidden field*, then choose *speed violations audit* as the field type.

**speed violations audit:** invisibly audio-record in response to a certain number of "speed violations" (cases where the enumerator spent less time on fields than specified in the *minimum\_seconds* column; this feature works in the Android app only since web browsers don't allow invisible audio recording). Enter **v=#; d=#** into the *appearance* column, where the first # is the number of violations that will trigger the audio recording, and the second # is the number of seconds of audio to record (e.g., **v=5; d=120** to begin recording after the fifth violation and record two minutes of audio). To learn more about speed violations, see the help topic on collecting high-quality data.

In the spreadsheet form definition:

type	name	appearance
speed violations audit	fieldname	v=#; d=#

In the form designer, click + *Add hidden field*, then choose *speed violations count* as the field type.

**speed violations count:** invisibly record the number of "speed violations" (cases where the enumerator spent less time on fields than specified in the *minimum\_seconds* column). To learn more about speed violations, see the help topic on collecting high-quality data.

In the spreadsheet form definition:

type	name
speed violations count	fieldname

In the form designer, click + *Add hidden field*, then choose *speed violations list* as the field type.

**speed violations list:** invisibly record the list of all fields for which "speed violations" occur (cases where the enumerator spent less time on fields than specified in the *minimum\_seconds* column). The list of fields will be space-separated. To learn more about speed violations, see the help topic on collecting high-quality data.

In the spreadsheet form definition:

type	name
speed violations list	fieldname

Tip: if you monitor incoming data with SurveyCTO's built-in *Data Explorer*, the submission-details view will use your speed-limit count and list data to visually flag speed-limit violations.

## Field types: text audit

In the form designer, click + *Add hidden field*, then choose *text audit* as the field type.

**text audit:** record meta-data about survey administration. If you want to save meta-data for only a proportion of forms, enter **p=#** into the *appearance* column (e.g., **p=50** to save meta-data for a random 50% of filled-out forms). To learn more about auditing and the format of the saved meta-data, see the help topic on collecting high-quality data.

In the spreadsheet form definition:

type	name	appearance
text audit	fieldname	
text audit	fieldname	p=#

Tip: if you monitor incoming data with SurveyCTO's built-in *Data Explorer*, you can easily download text-audit data – and even click the hourglass button when viewing a submission with text-audit data attached, to see the timing information overlaid on the submission-detail view.

## Field types: audio audit

In the form designer, click + *Add hidden field*, then choose *audio audit* as the field type.

**audio audit:** audio-record some or all survey administration (invisibly). If you don't specify any options in the *appearance* column, the first five minutes of every survey will be recorded. However, you have a range of options available:

- First, if you want to perform the audio audit on only a proportion of forms, enter **p=#** into the *appearance* column (e.g., **p=50** to audit a random 50% of forms).
- Then, for those audits that do take place, you can indicate alternative starting points and alternative recording durations. To start recording a certain number of seconds into a survey, enter **s=#** into the *appearance* column (where # is the number of seconds into the survey when recording will begin). To start recording at a random point, enter **s=#-#** instead, where #-# specifies a range within which to randomly select a start time (again, in seconds). To record for a maximum duration other than 300 seconds, enter **d=#** into the *appearance* column (where # is the maximum number of seconds to record; recording will stop after the assigned number of seconds ends or at the end of the survey, whichever comes first).
- To include multiple parameters in the *appearance* column, separate them with a semicolon [;] (for example, **p=10;s=0-1800;d=120** will record 10% of surveys, starting at a random point within the first 30 minutes of the survey, and lasting for 2 minutes).
- Alternatively, you can begin and end recording at specific questions. Specify **s=startfield** to begin recording at the field named "startfield" (i.e., when the user first swipes to the startfield question) and **d=endfield** to stop recording at the end of the field named "endfield" (i.e., when the user first swipes away from the endfield question). For example, **p=10;s=resp\_name;d=resp\_age** will record from the beginning of the resp\_name question to the end of the resp\_age question – but only for a random 10% of surveys.

Please note, however, that web browsers do not support invisible audio recording, so audio audits only work when users complete forms on a phone or tablet using the Android app.

To learn more about auditing, please see the help topic on collecting high-quality data.

In the spreadsheet form definition:

type	name	appearance
audio audit	fieldname	
audio audit	fieldname	p=#
audio audit	fieldname	p=#;s=#;d=#
audio audit	fieldname	p=#;s=#-#;d=#
audio audit	fieldname	p=#;s=startfield;d=endfield

Tip: if you monitor incoming data with SurveyCTO's built-in *Data Explorer*, you can easily download and play audio audits when looking at individual submissions.

#### Multiple audit fields in a single form

If you wish to have more than one audio audit field in a single form, please keep the following limitations and interactions in mind:

1. *You cannot include different types of audio audits in the same form.* That is, you cannot add both time-based audio audits and question-based audio audits in the same form. If you do, only the first time-based audio audit will work. The other audio audits will not be captured.
2. *If you have multiple time-based audits in the same form,* the form will treat them as mutually-exclusive and at most *one* of the audits will be captured. Which audio audit is captured (if any) will be determined by the probability you specify for each. For example, if you create two time-based audio audits, one with p=20 (that is, it should be captured for 20% of surveys) and one with p=30 (for 30% of surveys): in any given survey, the first audio audit will be captured if a new random draw is between 0 and 0.20, and the second audit will be captured if the random draw is greater than 0.20 and less than or equal to 0.50.

Each submission's random draw is a number between 0 and 1, representing probability ranges from 0% to 100%. If you specify p=100 for the first time-based audit in your form (that is, the audio audit should be recorded in 100% of surveys), then only that first audit will ever be triggered; no additional time-based audio audits added to your form will ever be captured.

3. *If you have multiple question-based audits in the same form,* they will *not* be treated as mutually-exclusive and all of them can be triggered and captured, even in the same submission. For example, if you create two question-based audits in your form and specify that one audit be captured from Q1 to Q10 with p=100 (that is, the audio should be captured in all surveys) and another audit be captured from Q15 to Q20, also with p=100: both question-based audio audits will be recorded in every submission.

However, the field ranges of these question-based audits should not overlap. If they do, only the first audio audit will be triggered and captured. For example, if you create two question-based audits, one that should be captured from Q1 to Q10 and a second one that should be captured from Q5 to Q15: only the first question-based audit will be captured.

## Learning from samples

Particularly when working directly with spreadsheet form definitions, it often helps to see – and to copy and paste from – examples. You can browse our library of sample forms [here](#).

To try a sample form out – or to use it as a starting-point for one of your own forms – just go to the Design tab, scroll down to the *Your forms and datasets* section, and click + then *Start new form*. You can then enable *Use a sample form as your starting point* and choose which sample to use.

## Other field properties

For any visible field, there are other properties that you can set. In the online form designer, you can easily edit these properties whenever adding or editing a field; in the spreadsheet form definition, they appear as columns on the *survey* worksheet:

- Text in the *hint* column will appear italicized under the field's label. (Unlike the *label* column, hints cannot include HTML formatting and are always shown as italicized.)
- The user's selection or entry will default according to the value in the *default* column, if any. Alternatively, you can use the *calculation* column to specify a dynamically-calculated default value. See the help topic on providing default entries or selections for fields for more details.
- Enter "yes" into the *required* column to require that a non-blank value be entered or selected before moving to the next question.
- Specify image, audio, or video filenames in the *media:image*, *media:audio*, and/or *media:video* columns to include visual or audio media as part of any question. If you're working directly with spreadsheet form definitions, then you must remember to always upload all media files when you upload the form to the SurveyCTO server (and the filenames for all uploaded files must *exactly* match the names you specify in the form definition itself). Which media file formats are supported will depend on your Android device, but all devices support common image formats like .jpg or .png. For an example, see the *images* sample form.

There are a few other columns on the *survey* worksheet that can be used, though they do not appear in the online form designer:

- Enter "yes" into the *disabled* column to temporarily disable a field. You might do this, for example, if you want to remove a question from a survey in the field, but want to keep it in your form definition for some reason. (If you want to remove a field for new submissions but keep exporting data for old submissions, put "0" in the *relevance* column instead. That will make it so the field never appears, but it will technically remain a part of the form and so existing data will still be exported for it.)
- Enter "yes" into the *read only* column to indicate that a field is read-only (i.e., the user cannot enter any value).

## Grouping and repeating questions



You can organize some or all of your questions into distinct groups. You might do this for internal-organization purposes, for more easily managing when whole groups of questions should appear, or for making your form easier to navigate.

In the form designer, click + *Add a group*, give it a label, and then click *Configure* to name and configure options for the group.

In the spreadsheet form definition, you can insert rows into your *survey* worksheet that mark the beginning and end of groups. The row that marks the beginning should have "begin group" in the *type* column, a short name for the group in the *name* column (without spaces or punctuation), and a label for the group in the *label* column. The row that marks the end of the group should just have "end group" in the *type* column and nothing else in any of the other columns.

Some options to keep in mind for any groups that you create:

1. You can specify a *relevance expression* to indicate that the group should only be displayed when appropriate (see *Implementing skip patterns with "relevance"* below).
2. You can repeat the questions in a group *n* times (much more on this below).
3. You can nest groups (put groups within groups) if you wish.
4. If you specify "organized" in the *appearance* column of a "begin group" row, the fields within the group will be visibly organized under the group's label whenever the group's fields are listed. This affects the list of fields that appears when you edit a saved form or use the *Go-to Prompt* option to jump to another field while filling out a survey form. Group labels will be colored red if any required fields within that group still need to be filled out, otherwise they will be colored green (this coloring of labels always happens in the web interface; in Android, it only happens if the *Color form navigation* setting is enabled in *SurveyCTO Collect*). See *Designing for easy navigation* for more on optimizing your form's navigation.
5. To display an entire group of questions on a single page, specify "field-list" in the *appearance* column of a "begin group" row. (You can also combine this appearance option with the above by specifying "organized field-list" in the *appearance* column.)
6. If you have a series of *select\_one* or *select\_multiple* fields that share the same option labels, you can put them all into a single "field-list" group and display them all in a compact table. The first *select\_one* or *select\_multiple* field in the series will provide column headers in the top row of the table: simply add "label" to the *appearance* column of this field. In the *select\_one* or *select\_multiple* fields that follow, add "list-nolabel" to the *appearance* column so that they appear as additional rows in the table.

For combining questions on a single page, it's often easiest to see (and copy from) a full example: see the sample form in *Field lists: Multiple questions on a single screen*.

In the spreadsheet form definition:

type	name	label	relevance	appearance
begin group	groupname	group label		
... fields ...				
end group				

begin group	groupname	group label		organized
... fields ...				
end group				
begin group	groupname	group label		field-list
... fields ...				
end group				

### Repeating questions

To ask a group of questions repeatedly, until the surveyor indicates that he or she is done, use a "repeat group." This works well, for example, for a series of questions about each household member.

In the form designer, click + *Add a group*, give it a label, and then say "yes" to *Repeat the questions in this group?*. In the spreadsheet form definition, use "begin repeat" and "end repeat" in lieu of "begin group" and "end group".

Note, however, that exported data for repeated groups is organized a bit differently because a single submission can have multiple answers to a single question – so it's not as simple as each submission having a single response for a repeated field that can be placed in a single row+column in your exported data. See *Understanding the format of exported data* for details on the "long" and "wide" formats supported by SurveyCTO.

To repeat a group of questions a fixed number of times based on a prior field value, include `${fieldname}` in the *repeat\_count* column (where "fieldname" is the name of a required *integer* field that contains the number of times to repeat the questions).

For example, you might have a field named "num\_hh\_members" that asks for the number of household members, then have a group of household roster questions with "`${num_hh_members}`" in the *repeat\_count* column of the "begin group" row. In that case, the survey would ask the household roster questions once for each member of the household.

For a full discussion of your options for repeated questions, see *Repeating fields (e.g., for household rosters)*. For working examples, see the *Rosters: two methods for repeated questions* and *Rosters: A third, hybrid method for repeated questions (repeat\_count)* sample forms.

Finally, to make it easier for users filling out repeat groups to keep track of where they are, review data entered so far, and navigate around, you can put "table" or "table-labeled" in the *appearance* column of your "begin repeat" rows. If you do, a table will appear underneath each repeated question and will update automatically as users provide their responses. It will allow them to easily review data, and they can click cells in the table to jump elsewhere within the repeat group. The only difference between "table" and "table-labeled" is how rows are labeled: "table" simply numbers rows while "table-labeled" also includes the repeat group's label (which can include `${fieldname}` references to include fields from within the repeat group itself). Tip: since the table's header row will include your field names, you should try to name your fields with friendly, descriptive names whenever you use one of these appearances.

In the spreadsheet form definition:

type	name	label	appearance	repeat_count
begin repeat	groupname	group label		
... fields ...				
end repeat				
begin repeat	groupname	group label	table	
... fields ...				
end repeat				
begin repeat	groupname	\${fieldname}	table-labeled	
calculate	fieldname			
... other fields ...				
end repeat				
begin repeat	groupname	group label		\${fieldname}
... fields ...				
end repeat				

## Using constraints to validate responses

Fields in your survey can either be *required* – in which case they require a non-blank response from the user – or not required. And, for any non-blank response that is entered or selected by the user, you can use a *constraint* to validate the response; if the user's entry or selection does not meet your constraint, the user will have to correct it before continuing.

In the form designer, you can control the *required* and *constraint* settings whenever you add or edit a field. And when adding a constraint, there is a step-by-step wizard that will help you to add common types of field validation.

In the spreadsheet form definition, the *required* column of your *survey* worksheet will determine which fields the user can leave blank and which must be filled; enter "yes" into that column to require a response. Then, enter an expression into the *constraint* column to further validate whatever response is entered or selected by the user.

If the user does enter or select a response, they will only be allowed to proceed to the next question when the expression in your *constraint* column evaluates to true. If the user tries to move forward with a non-blank response but the expression is false, then a generic "That entry is invalid" message will appear; to override this message on a field-by-field basis, enter another message into the *constraint message* column. (This is the default manner in which constraints are enforced. However, in the *General settings* for the *SurveyCTO*

*Collect* app, you can change the constraint-processing option to "Defer validation until finalized" to skip constraint enforcement until the very end, when users try to finalize their forms for submission. For most forms, though, the default behavior is what you want.)

For example, ". < 130" requires that the entered value be less than 130; ". < 130 and . > 10" also requires it to be greater than 10. You might couple such constraints with a *constraint message* like "Please enter a valid age." In these constraints, the "." represents the current entry or selection.

You can also compare with a prior field's value, as in ". < 130 and . > 10 and . > \${daughterage}".

The easiest way to build a constraint expression is to use the wizard built into the form designer, or to use the constraint-builder on the server's Design tab: just select *Tools*, then *Build constraint* from the *Your forms and datasets* section. You can even use the constraint-builder offline: simply run *SurveyCTO Sync* and select the constraint-builder from the *Tools* menu's *Form tools* sub-menu.

For details on hand-creating or hand-editing constraint expressions, see the *Using expressions in your forms* section below.

To easily test a constraint without having to fill out your actual form, click the *Tools* icon, then *Test constraint* from the *Your forms and datasets* section of the Design tab. To use the constraint-tester offline, simply run *SurveyCTO Sync* and select the constraint-tester from the *Tools* menu's *Form tools* sub-menu.

## Implementing skip patterns with "relevance"

Fields and groups in your survey can be shown – or not – depending on the conditions you set for their *relevance* (i.e., when they should be skipped vs. when they should be considered relevant).

In the form designer, you can control the *relevance* settings whenever you add or edit a field or group. There is a built-in wizard that will help you to add one or more common conditions (for example, "the gender field is equal to the text value 'F'").

In the spreadsheet form definition, you can enter an expression into the *relevance* column to control when the field or group should display. It will then show up to users only when that expression evaluates to true; otherwise, it will remain hidden (and its response will be empty).

Typically, whether a question displays will depend on the responses to prior questions. For example, you might ask for informed consent at the beginning of your survey. Then, for all remaining questions, you might want to include "\${consent} = 1" in the *relevance* column. That way, all remaining questions are automatically skipped when consent is not given. Another approach is to enclose all consent-requiring questions within "begin group" and "end group" rows, then specify the relevance expression on the "begin group" row – so you only have to enter the expression once (see the *Grouping and repeating questions* section above).

Of course, there may be multiple criteria for displaying a certain question. For example, the relevance expression for a question about the fourth household member might look like "\${consent} = 1 and \${nhhmembers} > 3".

The easiest way to build a relevance expression is to use the wizard built into the form designer, or to use the relevance-builder on the server's Design tab: select *Tools*, then *Build relevance* in the *Your forms* section. You can even use the relevance-builder offline: simply run *SurveyCTO Sync* and select the relevance-builder from the *Tools* menu's *Form tools* sub-menu.

For details on hand-creating or hand-editing relevance expressions, see the *Using expressions in your forms* section below.

Please note that all relevance expressions are only evaluated once, when a screen is first displayed. Thus, if you use a "field-list group" to display multiple questions on one screen, one question cannot appear or disappear based on the answer to another question that appears above it on the same screen. To implement skip patterns, questions must appear on separate screens.

## Using expressions in your forms

In the form designer, you can build constraint, relevance, and calculation expressions using simple wizards that are built right into the designer. You can also edit expressions by hand, in which case the reference materials below will come in handy.

Even when you're editing spreadsheet form definitions directly, you can use the constraint-builder, the relevance-builder, and/or the calculation-builder, all of which are available both as actions in the *Your forms* section of your server's Design tab and, offline, as options in the *Tools... Form tools* menu of *SurveyCTO Sync*.

If you want to create or edit expressions by hand, you can refer to user responses and other fields as follows:

- For constraints, "." is used to refer to the user's proposed entry or selection for the current field (i.e., for the value you are testing to see if it's valid).
- Use "\${fieldname}" to refer to a prior field's entry, selection, or calculated value. (This gives the value exactly as it will later appear in your data.)

And you can use all of the following operators in any expression:

Operation	Operator	Example	Example answer
Addition	+	1 + 1	2
Subtraction	-	3 - 2	1
Multiplication	*	3 * 2	6
Division	div	10 div 2	5
Modulus	mod	9 mod 2	1
Equal	=	\${fieldname} = 3	true or false
Not equal	!=	\${fieldname} != 3	true or false
Greater-than	>	\${fieldname} > 3	true or false
>-or-equal	>=	\${fieldname} >= 3	true or false
Less-than	<	\${fieldname} < 3	true or false
<-or-equal	<=	\${fieldname} <= 3	true or false

Or	or	<code>\${fieldname} = 3 or \${fieldname} = 4</code>	true or false
And	and	<code>\${fieldname} &gt; 3 and \${fieldname} &lt; 5</code>	true or false
Not	not()	<code>not(\${fieldname} &gt; 3 and \${fieldname} &lt; 5)</code>	false or true

You can also call any of the following functions:

1. **string-length(field)**: Returns the length of the string field (as in the constraint "string-length(.) > 3 and string-length(.) < 10").
2. **count-selected(field)**: Returns the number of items selected in a *select\_multiple* field (as in the constraint "count-selected(.) = 3").
3. **selected(field, value)**: Returns true or false depending on whether the value indicated in the second parameter was selected in the *select\_one* or *select\_multiple* field indicated in the first parameter (as in the relevance "selected(\${fieldname}, 'Male')"). Please note that the second parameter to the selected() function should always specify the internal value associated with the selection, as configured in the *choices* worksheet's *value* column. (You cannot specify the label configured in the *label* column.)
4. **selected-at(field, number)**: When the passed number is 0, returns the first selected item in a *select\_multiple* field; when the passed number is 1, returns the second selected item; etc. (as in the relevance "selected-at(\${fieldname}, 0) = 'Shona'"). Note that the returned value will correspond with the internal value associated with the selection, as configured in the *choices* worksheet's *value* column.
5. **jr:choice-name(value, 'field')**: Returns the label for a *select\_one* or *select\_multiple* field choice (as in the calculate expression "jr:choice-name(\${selectonefield}, '\${selectonefield}')" to return the label for the currently-selected choice in the field named "selectonefield"). For *select\_multiple* fields, you can combine with selected-at() to get the label for individual selections (as in "jr:choice-name(selected-at(\${selectmultifield}, 0), '\${selectmultifield}')" to return the label for the first choice in the field named "selectmultifield").
6. **concat(fieldorstring, fieldorstring, ...)**: Concatenates fields (and/or strings) together (as in the calculate expression "concat(\${firstname}, ' ', \${lastname})").
7. **duration()**: Returns the total amount of time spent, in seconds, filling or editing the current form submission. Call this function in a *calculate\_here* field to capture the user's number of seconds into a form when they first reach a particular point (expression for the *calculation* column: "once(duration())"); you can then subtract one captured duration from another to get the time spent in between. Or, use in a regular *calculate* field to capture the total duration spent on the form overall (expression for the *calculation* column: "duration()").
8. **count(repeatgroup)**: Returns the current number of times that a *repeat group* has repeated (i.e., the number of "instances" of the group, as in the calculate expression "count(\${repeatgroupname})").
9. **sum(repeatedfield)**: For a field within a *repeat group*, calculates the sum of all values (as in the calculate expression "sum(\${loan\_size})").
10. **join(string, repeatedfield)**: For a field within a *repeat group*, generates a string-separated list of values (as in the calculate expression "join(' ', \${hh\_member\_name})" to generate a single comma-separated list from all entered names).

11. **min(repeatedfield):** For a field within a *repeat group*, calculates the minimum of all values (as in the calculate expression "min(\${hh\_member\_age})"). If more than one non-repeating field is passed to min(), then the minimum of the passed fields will be returned (as in "min(\${father\_age}, \${mother\_age})").
12. **max(repeatedfield):** For a field within a *repeat group*, calculates the maximum of all values (as in the calculate expression "max(\${hh\_member\_age})"). If more than one non-repeating field is passed to max(), then the maximum of the passed fields will be returned (as in "max(\${son\_age}, \${daughter\_age})").
13. **index():** Called within a *repeat group*, returns the index number for the current group or instance (as in the calculate expression "index()", which will return 1 for the first instance of the repeat, 2 for the second, and so on). (We used to recommend the ODK position(..) function instead of index() – but that would fail if you called it from within a non-repeating group that was itself within a repeating group.)
14. **once(expression):** For use in expressions in *calculate* or *calculate\_here* fields only, indicates that the enclosed expression should be calculated only once per form (as in the expression "once(random())" to draw a random number). If a calculated expression is not enclosed in once(), it will recalculate periodically, including each time that the form is edited and saved. In the case of *random()*, this would draw a new random number every time the form is edited and saved – which is not generally what you want. *Be sure to only use once() on the very outside of an expression* (so "once(random()\*2)" is okay, "2\*once(random())" is not); to avoid potential problems, use "once(...)" alone in one *calculate* field, then you can use that field in more complex expressions elsewhere.
15. **once(random()):** Returns a random number between 0 and 1 (as in the calculate expression "once(random())"). To introduce randomness into your forms, use once(random()) in calculated fields, then refer to those calculated fields in your other expressions (don't nest once(random()) directly inside another function or expression). You should always call random() inside the once() function and you should never use random() directly in a relevance expression, because you don't want to generate a new random number every time the relevance is calculated: you want just one stable random number for each filled-out form. For an example, see the sample form in *Randomization: Randomizing form elements*.
16. **pulldata(source, colname, lookupcolname, lookupval):** Pulls data from a dataset or .csv file (as in the calculate expression "pulldata('hhplotdata', 'plot1size', 'hhid\_key', \${hhid})", which will pull a value either from an attached .csv file named *hhplotdata.csv* or from an attached dataset with the unique ID *hhplotdata*; the value will come from the *plot1size* column of the pre-loaded data, and the *hhid* field will be used to identify the matching row in the pre-loaded data's *hhid\_key* column). See *Preloading data into a form* for more information, or *Pre-loading: Referencing pre-loaded .csv data* for a working example.
17. **indexed-repeat(repeatedfield, repeatgroup, index):** References a field or group that is inside a prior *repeat group*. The first parameter specifies the prior field or group in which you are interested; the second specifies the prior repeat group within which the field or group of interest is located; and the third specifies the instance number, within the prior repeat group, to use. For example, the calculate expression "indexed-repeat(\${name}, \${names}, 1)" will return the first name available when the "name" field is inside a prior repeat group named "names". From inside a later repeat group, the calculate expression "indexed-repeat(\${name}, \${names}, index())" will pull the xth name from the prior repeat group, where x is the instance number of the current repeat group (e.g., if currently in the fourth instance of a repeat group, it will return the fourth name from the earlier repeat group). If you need to reference a field or group within multiple nested repeat groups, you can supply additional parameters to indicate the instance numbers to use for each level of nesting. For example, the calculate

expression "indexed-repeat({name}, {families}, {familynumber}, {names}, {membernumber})" will pull a particular family member's name when family member names are inside a repeat group that is itself inside a repeat group of families. For two examples, see *Rosters: Two methods for repeated questions*. Finally, note that when the passed-in instance number is invalid, an instance number of 1 will be automatically used instead (so the first instance will be returned for such cases).

18. **rank-index(index, repeatedfield)**: calculates the ordinal rank of the specified instance of a repeated field for use outside the repeat group (as in the calculate expression "rank-index(1, {random\_draw})" to calculate the rank of the first instance, based on the value of its "random\_draw" field as compared with other instances' values). The rank of 1 is assigned to the instance with the highest value, the rank of 2 to the instance with the next-highest value, and so on. Instances with the same value are ordered arbitrarily (they are not given the same rank, so every instance will have a unique rank). If you pass an invalid index or an index to an instance with a non-numeric value, a rank of 999 will be returned. See below for a variation that can be used within the same repeat group.
19. **rank-value(fieldorvalue, fieldorlist)**: calculates the ordinal rank of a given value relative to a list of values (as in the calculate expression "rank-value(3, '4 2 1 9 3 7')" to calculate the rank of 3 in the given list, or as in "rank-value({random\_draw}, {list\_of\_draws})" if the value and the list are both stored in fields). The rank of 1 is assigned to the highest value in the list, the rank of 2 to the next-highest value, and so on. If you pass an empty list or a value that's not in the list, a rank of 999 will be returned. Use this variation to calculate the rank within the same repeat group as the field being ranked. In this use case, add a *calculate* field outside the repeat group to construct the full list of values using *join()* (as in "join(' ', {random\_draw})" when the name of the repeated field is "random\_draw"); then, within the repeat group, find the current instance's rank with another *calculate* field (as in "rank-value({random\_draw}, {list\_of\_draws})" when the name of the joined list you added outside the repeat is "list\_of\_draws").
20. **de-duplicate(string, field)**: For a string-separated list of items, removes duplicates (as in the calculate expression "de-duplicate(' ', {fieldname})" to remove duplicates from a space-separated list). For example, say you had a repeated multiple-choice field. Outside the repeat group itself, you might want to join all selections together into one calculated list, as in "join(' ', {repeatedfield})" – but that list might then contain the same selections multiple times. The calculate expression "de-duplicate(' ', join(' ', {repeatedfield}))" would join together all of the selections, and remove duplicates. You could then call *count-selected()*, *selected-at()*, etc. on the combined list in the calculated field.
21. **distance-between(geopointfield1, geopointfield2)**: Returns the distance, in meters, between two *geopoint* fields (as in the calculate expression "distance-between({start\_gps}, {end\_gps})"). (Keep in mind that the accuracy of the distance calculated will depend on the accuracy of the GPS readings, so try to be sure to get accurate GPS readings.)
22. **area(repeatedgeopointfield)**: Returns the area enclosed, in square-meters, within a series of repeated *geopoint* fields (as in the calculate expression "area({gps\_reading})", called outside a *repeat group* that includes a field named "gps\_reading"). (Keep in mind that the accuracy of the area calculated will depend on the accuracy of the GPS readings, so try to be sure to get accurate GPS readings.)
23. **short-geopoint(geopointfield)**: Returns a string containing the GPS location with only the longitude and latitude, no altitude or accuracy (as in the calculate expression "short-geopoint({location})"). You might use this if publishing data to outside systems like Google Fusion Tables, which might be confused by altitude or accuracy.



24. **substr(fieldorstring, startindex, endindex)**: Returns a substring starting at startindex and ending just before endindex (as in "substr(\${phone}, 0, 3)" to get the first three digits of a phone number). Indexes start at 0 for the first character in the string.
25. **coalesce(field1, field2)**: Returns field1 if it isn't empty, otherwise returns field2 (as in the calculate expression "coalesce(\${id}, \${id2})").
26. **round(field, digits)**: Rounds the numeric field value to the specified number of digits after the decimal place (as in the calculate expression "round(\${interest\_rate}, 2)").
27. **regex(field, expression)**: Returns true or false depending on whether the field matches the regular expression specified (as in the constraint "regex(., '[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}')" which checks for a valid-looking email address). Because regular expressions can be arbitrarily complex, this function allows for construction of advanced relevance and constraint expressions.
28. **if(expression, valueiftrue, valueiffalse)**: Returns one of two values, depending on whether an expression is true (as in the calculate expression "if(selected(\${country}, 'South Africa') or selected(\${country}, 'Zimbabwe'), 'SADC', 'Non-SADC')").
29. **format-number(field)**: Formats an *integer* or *decimal* field according to the user's locale settings (as in the calculate expression "format-number(\${income})", which might format "120000" as "120,000").
30. **number(field)**: Converts field to a number (as in "number('34.8') = 34.8").
31. **int(field)**: Converts field to an integer (as in "int('39') = 39").
32. **string(field)**: Converts field to a string (as in "string(34.8) = '34.8'").
33. **date(string)**: Converts string into a date (as in the relevance "\${fieldname} > date('2013-01-31')").
34. **date-time(string)**: Converts string into a date-time (as in the relevance "\${fieldname} > date-time('2013-01-31T16:42:00')").
35. **format-date-time(field, format)**: Converts date and/or time into a string (as in the calculate expression "format-date-time(\${fieldname}, '%Y-%b-%e %H:%M:%S')"). In the format string, %Y indicates four-digit year, %y two-digit year, %m two-digit month, %n one-or-two-digit month, %b three-letter month, %d two-digit day, %e one-or-two-digit day, %H two-digit hour, %h one-or-two-digit hour, %M two-digit minute, %S two-digit seconds, %3 three-digit milliseconds, and %a three-letter day of week.
36. **today()**: Returns the current date (as in the calculate expression "format-date-time(today(), '%Y-%b-%e')").
37. **now()**: Returns the current date and time (as in the *calculate\_here* expression "once(format-date-time(now(), '%Y-%b-%e %H:%M:%S'))" for saving the date and time at which a particular point in your form is first reached).
38. **uuid()**: Calculates a unique random ID (as in the calculate expression "once(uuid())").
39. **version()**: Returns the version number of the current form (as in the calculate expression "version()").
40. **username()**: Returns the currently-configured username of the user filling in the form (as in the calculate expression "username()").
41. **linebreak()**: Returns a linebreak character (as in the calculate expression "concat(\${field1}, linebreak(), \${field2}, linebreak(), \${field3})" for storing a list of three field values with linebreaks between them).

42. **hash(fieldorvalue, ...)**: Returns a hash value that represents the one or more parameters passed (as in the calculate expression "hash({name})" or "hash({name}, {birthday})"). See this page for more details on hash values and their potential uses.
43. **pow(base, exponent)**: Returns the field, number, or expression in the first parameter raised to the power of the field, number, or expression in the second parameter (as in the calculate expression "pow(1+{annual\_interest\_rate}, {years\_of\_interest})").
44. **log10(fieldorvalue)**: Returns the base-ten logarithm of the field or value passed in.
45. **sin(fieldorvalue)**: Returns the sine of the field or value passed in, expressed in radians.
46. **cos(fieldorvalue)**: Returns the cosine of the field or value passed in, expressed in radians.
47. **tan(fieldorvalue)**: Returns the tangent of the field or value passed in, expressed in radians.
48. **asin(fieldorvalue)**: Returns the arc sine of the field or value passed in, expressed in radians.
49. **acos(fieldorvalue)**: Returns the arc cosine of the field or value passed in, expressed in radians.
50. **atan(fieldorvalue)**: Returns the arc tangent of the field or value passed in, expressed in radians.
51. **atan2(x, y)**: Returns the angle in radians subtended at the origin by the point on a plane with coordinates (x, y) and the positive x-axis, the result being in the range -pi() to pi().
52. **sqrt(fieldorvalue)**: Returns the non-negative square root of the field or value passed in.
53. **exp(x)**: Returns the value of  $e^x$ .
54. **pi()**: Returns pi.

*Please note that literal strings within your expressions should always be enclosed in single-quotes, as in 'Male' or ' ' for a single space. If you use double-quotes (like "Male" or " "), your expressions will not work properly.*

## Updating an existing form

Even after your form has been deployed and users have started collecting data, you can safely make and deploy many changes. You just want to be careful to understand the limitations and implications discussed below.

In the form designer, you can update an existing form by simply making changes and then clicking on *Save* (to save as a draft) or *Save and deploy* (to save and deploy for actual data-collection). Whenever you edit a form that has already been deployed, you should consider the implications discussed below. In particular, you should be careful about removing fields, renaming fields, or changing groups.

To update an existing form by editing the spreadsheet form definition directly:

1. **Update the form definition's survey and choices worksheets as desired.**
2. **Increase the version number on the settings worksheet.** If you started with a SurveyCTO form template or with one of the sample forms, then this is automatic: the version is set to a formula that automatically increments every minute. (Otherwise, you need to increment the version number yourself, taking care to keep the number of digits fixed: the version has to be a single whole number, and its number of digits has to stay the same from one version to the next. The convention is to use a ten-digit number that represents the current date and time, as in 1401130917 for 9:17AM on January

13, 2014. Alternatively, you can use 10-digit numbers like 0000000001 and 0000000024 or 3-digit numbers like 001 and 024.)

3. **Upload the new version of the form definition to your SurveyCTO server.** To do this, navigate to the Design tab, scroll down to the appropriate form in the *Your forms and datasets* list, and click the *Upload* button. (As part of your upload, be sure to include any necessary media files referenced by your form, and/or check the box to *Keep old attachments*.)

However you update your form, your next step will be to deploy the updated form to your users. If you're collecting data via the web, this will happen automatically: whenever a web user begins a new submission, they will always use the latest version of your form. Your Android users who use *SurveyCTO Collect*, on the other hand, can click *Get Blank Form* to manually download the new version from your server. They can also configure *Collect* to auto-download new form versions whenever they are available, by going into *Collect's General settings* and checking *Auto download with Wi-Fi* and/or *Auto download with network* ("network" here refers to "cellular network"). When either of these options is enabled and the appropriate connection type is available, *Collect* will check hourly for form updates; available updates will then be downloaded automatically, but they won't be installed until the user confirms the installation by clicking on the *Install Form Updates* button that appears, or by confirming the update when about to fill out an older version of the form.

*SurveyCTO Collect* – on Android or on the web – will always present the latest installed version of your form when a user selects *Fill Blank Form*, but older versions of your form are kept to support users who had already begun submissions with those earlier versions. Whether a user was in the middle of editing an old version of the form when you updated it, or they came back much later to *Edit Saved Form* a form begun with an old version, existing submissions will not be affected by any of your changes. Only new submissions, begun *after* you updated your form, will benefit from the changes you made.

*We strongly encourage you to maintain back-ups of all deployed versions of your forms, and to keep detailed notes on the changes you make.* You can download the spreadsheet form definition for any form by clicking *Export* in the designer or *Download* in the *Your forms and datasets* section of the Design tab. If you're editing spreadsheet form definitions directly, one easy strategy is to add a "notes" worksheet to each of your survey forms, and to add detailed notes there as you make changes; *SurveyCTO* will simply ignore the extra worksheet, but it can be incredibly valuable to your team – particularly as they work to interpret collected data.

#### Implications for your data

*SurveyCTO* keeps track of each form based on its unique form ID (on the *settings* worksheet), so two forms uploaded with the same form ID but different version numbers are, by definition, two versions of the same form, and two forms uploaded with different form IDs are, on the other hand, treated as two different forms. Ultimately, all data is stored, aggregated, viewed, and exported based on the form ID, so the data for different form versions that share a single form ID will mix together.

When viewing or exporting your data, the view or export will be based on the current (latest) deployed version of the form available on your server. That version of the form is what will govern which fields and which groups of fields get displayed or exported. Data collected with earlier versions of the form may be missing some data (in the case of fields or groups that were added later) and may have some data for fields or groups that have since been deleted. In the former case, where fields are missing for a submission, those fields will be exported as blank or empty – unless you set the preferences option in *SurveyCTO Sync* to specify some other character or string to use for missing values. In the latter case, where data exists for fields or groups that no longer exist in the latest version of the form, that data will simply not be exported at

all. (If you want to keep exporting data for fields that you would like to delete, don't actually delete those fields: instead, put the number 0 in their *relevance* column. That will keep the fields in the form so that they will be exported if present, but they will never show up to users filling out the latest version of your form.)

To help you in interpreting your data, every export will automatically include a *formdef\_version* column. For each row (i.e., for each submission), that column will contain the version number of the form definition used to fill out the form.

*You will want to take some care in how you alter the groups within which your form fields reside, because SurveyCTO internally tracks each field by a full name that includes all enclosing groups.* For example, a field named "age" within a group named "demographics" that is itself within a group named "module1" will be tracked as `"/module1/demographics/age"`. If we were to collect some data for that field and then decide to shorten the name of the "demographics" group to just "demo", then we would subtly alter our form with respect to the "age" field: old form submissions would have a field named `"/module1/demographics/age"` and new submissions would start having a field named `"/module1/demo/age"` instead. Exports would, by default, include only the `"/module1/demo/age"` field; data collected for the earlier `"/module1/demographics/age"` field would *not* be included. Now, if we set the preference to "Ignore groups so fields with same name export together" in *SurveyCTO Sync*, then our exports *would* include merged data for both "age" fields. But that kind of merging slows down the export process, and it can't work when you change anything about the *repeat groups* that enclose a field. So your best bet is to avoid adding or removing groups, renaming groups, or moving fields across groups.

### Changes to encryption settings

While you can change most things about an existing form, you cannot change the encryption settings. In other words, you can't change a form from unencrypted to encrypted, you can't change a form from encrypted to unencrypted, and you can't change the encryption key used to encrypt the form. To make any changes to a form's encryption settings, you will need to change the form's unique form ID on the *settings* worksheet so that the updated form will essentially be treated as a new form altogether. (This is because you simply cannot mix data for two form versions if those versions have different encryption settings. It would just never work.)

## Missing data from old form versions

Because the current (latest) version of a form governs how data is formatted and exported, it's easy to end up in a situation where data collected with an older version of a form doesn't display in the *Data Explorer* and doesn't appear in exports. The data exists – it hasn't been lost! – but it's not showing up when you look at or export a submission, so it certainly seems lost.

- Take a simple example: you delete a field and deploy an updated version of the form. If you had collected data with an older version of the form, then you might have data for that field. However, since the latest version of the form doesn't include that field any more, when you review an older submission in the *Data Explorer* or export your data, the field doesn't appear. SurveyCTO has data for that field, but there's no field in which to show it when you review the submission, and there's no column in which to export it when you export your data. So you don't see that data.
- Take another example: you rename a field. From SurveyCTO's standpoint, a renamed field is really just one field that was deleted and another one that got added. So if the current form collects data for a field named "age", then that's the field that shows up when you review submissions and the column

that appears in your export; if you have data for an earlier field named "childage", but no such field now exists in the form, then that data won't show up anywhere.

- A far more subtle case concerns changes to the groups within which fields are located: SurveyCTO keeps track of fields a little like computers keep track of files, using not only the field name, but also the groups within which the field resides. So a field named "age" in groups named "demographics" and "youngestchild" is tracked, internally, as /demographics/youngestchild/age. In fact, that field is treated as a different field from /demographics/oldestchild/age, even though both fields are named "age". If you add, remove, or rename groups, it implicitly renames all fields within those groups, which has the effect of making it like some fields have gone away and some new ones have been added. This can then have the net effect that data from older forms disappears from the *Data Explorer* and from data exports.

Please see the help topic immediately above this one for details on how to update existing forms, and for more on the implications of different kinds of changes to your form. If you need more help, you can post a message or open a request in our online support center.