

Collecting high-quality data

Collecting high-quality data can be hard – much harder, at least, than collecting poor-quality data. It involves thought and effort at several key stages of the data-collection process: survey design, training, and supervision and monitoring during data collection. Luckily, SurveyCTO has been designed to make your life systematically easier, so that you can collect higher-quality data with relatively less effort. Following are a series of quality-assurance-related suggestions, along with links to more details about particular SurveyCTO features that can help.

Survey design

Whether you're collecting data with an electronic device or paper and pencil, many of the same rules apply: you want to think hard about which questions you ask, how you ask them, and in what order; you want to exhaustively pre-test and pilot every question; and you want to make sure that everything you do gets translated properly (and re-tested) in all of the necessary languages or dialects. With electronic data-collection, you have some additional opportunities, however, to improve data quality via good design.

Skipping irrelevant questions

The first opportunity may seem obvious, but: don't ask questions that aren't relevant. By [automatically skipping irrelevant questions](#), you minimize the possibility of skip-pattern errors. And, while it is tempting to let your enumerators sort out which questions should be asked when, it is often better to design your survey so that *you* get to decide in a more predictable, error-free fashion. See [the help on relevance](#) for more on implementing automatic skip patterns.

Constraining responses

The next opportunity has to do with [data validation](#) as the enumerator is entering answers into your survey form: don't let the enumerator continue if the data is clearly incorrect, invalid, or inconsistent. Here, as the survey designer, you can control the quality of the data being entered by enumerators. You get to set a higher bar. And while you don't want to restrict answers to the extent that enumerators are totally unable to enter unusual (but sometimes correct) values, you should disallow answers that are clearly impossible or those that contradict earlier responses. After all, the cheapest and safest possible point at which to correct a mistake is during the interview itself. See [the help on constraints](#) for more on constraining enumerator responses.

If you don't want to fully disallow some kinds of responses but still want to warn the enumerator that the responses are suspicious, you can always implement a warning or confirmation (or "soft" constraint): just add a note or yes-no confirmation that is only [relevant](#) (i.e., that only shows) if there is something that looks potentially incorrect, invalid, or inconsistent. That way, your concern, as a survey designer, is raised under certain suspicious circumstances, but the enumerator is ultimately able to either go back and correct something or just continue without making any changes.

Designing for workflow

Finally, SurveyCTO gives you additional options for controlling the workflow surrounding how survey forms are filled out: how your enumerators navigate the survey, when and how they save or review their work, when and how they finalize data and send it in. It's helpful to keep these options in mind as you design your survey to fit the circumstances your enumerators are likely to face as they administer your survey in the field. See [the help on workflows](#) for more on designing your form for optimal workflow. [INCLUDED BELOW]

Supervision and monitoring

Supervision and monitoring of survey operations happens at several levels. First, there is field-level supervision and monitoring by team leaders and field supervisors. Then, presuming that your field teams are periodically sending data to the SurveyCTO server, you can also monitor data as it comes in; this gives you incredibly valuable opportunities to manage the overall data-collection process in ways that help to assure a high level of data quality.

Field-level supervision

One commonly-used tool for field-level supervision is the "back-check." Supervisors – or specially-hired back-checkers – return to some random proportion of the households or facilities surveyed, confirm that they were visited by enumerators, and re-ask certain questions to compare to the original responses. But it can be cumbersome to set up the workflow for back checks: you have to obtain all original surveys, select a certain percentage to be back-checked, compile and upload the original data from the selected sample to a back-checker's device, and then send them out to conduct the back-check. Using [datasets](#), you can automate the workflow for your back-check surveys and make the overall turn-around time much faster with less work. See [the back-checking sample](#) for a working example.

Office-level monitoring

The surest single way to improve the quality of your entire data-collection effort is to review your data as it comes in. You can [export all of your data](#), and you can also [export different subsets of your data](#) for review by different people or different teams. If you're collecting GPS locations, you can even [export to Google Earth](#) in order to review in a more visual way. And if reviewing lots of raw data in spreadsheet format is too difficult, you can

also [use Microsoft Word's mail merge feature](#) to create more easy-to-review versions of incoming data.

Meanwhile, you need to track your back-office data review, processing, and management processes. Because many people track these processes in spreadsheets, SurveyCTO makes it easy to merge (subsets of) incoming data with [Microsoft Excel workbooks](#) or [Google Spreadsheets](#).

You shouldn't have to manually catch every possible problem yourself, however. SurveyCTO can do a lot of that work for you: it includes a simple but sophisticated system for [configuring automated quality checks](#). Once appropriate quality checks have been configured, potential problems are detected automatically and your team is given specific warnings to help support their quality-assurance efforts. See [the help topic on quality checks](#) for more details.

Finally, you can [automatically execute other automated processes](#) whenever new data exports, in case you have built other types of automated processes for processing and/or checking incoming data.

Random audits

To help assure quality in your data-collection efforts, you can always have supervisors or other quality-assurance personnel randomly (a) accompany your surveyors and (b) re-visit surveyed individuals to perform back-checks. As an alternative or complement to these manual QA methods, SurveyCTO offers two random auditing options to allow you to monitor for quality survey administration.

The first auditing option is a random "text audit." For any random proportion of administered surveys (from 1% to 100%), SurveyCTO can save, along side the survey data itself, meta-data about survey administration. This includes details on how much time the surveyor spent on each question in the survey form, and on the sequence taken through the survey. For each audited survey, a .csv file is saved that contains this information. Once the survey data has been exported, these .csv files can be opened and reviewed in Excel. For example:

Field name	Total duration (seconds)	First appeared (seconds into survey)
intronote	3	0
consent	5	3
consented[1]/name	3	8
consented[1]/age	3	10
consented[1]/confirmnote	1	13

The second auditing option is a random "audio audit." For any random proportion of surveys, SurveyCTO can audio-record some or all of the survey administration. When the data is exported, it will include audio files for any audited surveys; QA personnel in the office can then review these audio recordings.

Random audio audits can be configured to begin at the beginning of a survey, at the beginning of a particular question, a fixed number of seconds into a survey, or a random number of seconds into a survey. They can be configured to record for a fixed duration (in seconds), or stop at the end of a particular question.

Each second of audio recording adds between 700 and 1,000 bytes to the survey data, so most configurations audit only a random portion of a random subset of administered surveys. Since surveyors are unaware of when they are being recorded, they cannot behave systematically differently when being audited.

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

To add auditing to any survey, add [text audit](#) and/or [audio audit](#) fields to the survey's form definition, name the fields, then specify configuration parameters in the *appearance* column. See the [FAQ on form design](#) for specific information on configuring both field types.

When you export data that includes audit files, those files will be exported to the *media* subdirectory, and the full path and filename for each file will be included in the .csv output.

Text audit files will be named "TA_UUID.csv" where UUID is the unique *instanceID* for the corresponding survey form.

Audio audit files will be named differently depending on their configuration. For recordings that begin at a particular question, the filename will be "AA_UUID_FIELDNAME.3gpp" where UUID is the unique *instanceID* and FIELDNAME is the name of the field at which the recording began. For other recordings, the filename will be "AA_UUID_AFTER_#S.3gpp" where UUID is the unique *instanceID* and # is the number of seconds into the survey before the audio recording began (0 for recordings that started at the beginning of the survey).

[See the auditing sample form](#) for a basic example.

Speed limits

Another quality-control tool available in SurveyCTO is "speed limits": the *minimum_seconds* column in your survey form can specify a minimum number of seconds that enumerators should spend on any given field.

If you do specify a *minimum_seconds* value for a field, then the first time that field is shown for a given survey, SurveyCTO will (invisibly) keep track of how much time the enumerator spends before moving on to another question. If the enumerator spends less than the specified minimum time the first time he or she encounters the field, nothing will happen by default; you have a few choices for how you deal with these "speed limit violations":

1. Quietly keep track of the number of violations. You can do this by adding a new field to your survey form with [speed violations count](#) as its field type. This will be an invisible field that keeps a tally of how many times the enumerator violated the speed limit when filling out the form. You might track the number of violations and then [set up quality checks](#) to warn about individual cases with many violations or about enumerators or teams with significantly different levels of violations on average.
2. Quietly keep track of the list of violations. You can do this by adding a new field to your survey form with [speed violations list](#) as its field type. This will be an invisible field that keeps a list of all fields for which the enumerator violated the speed limit when filling out the form. Often, if you keep track of the count of violations you also want to keep track of the list – so that you can better follow up with enumerators or even change the speed limits when appropriate.
3. Trigger audio audits upon a certain number of violations (only in the Android app). Here, rather than audio recording randomly (as described above), you would begin audio recording in response to a certain number of speed limit violations – so that your team can hear what was going on when the enumerator was moving so quickly through the survey. Just add a new field to your survey form with [speed violations audit](#) as its field type, then put "v=x; d=y" into that field's *appearance* column to specify the number of violations required to trigger the audit (x) and the length of the audit in seconds (y). For example, to invisibly record a two-minute audio clip beginning immediately after the fifth violation, put "v=5; d=120" in the *appearance* column. When you export your data, all audit recordings will be included in the *media* subdirectory, and the .csv data will include the path and filename within the audit field itself (for each case where an audit was triggered).
4. Enforce the speed limit (only in the Android app). Another option is to prevent the enumerator from moving forward until after the minimum time has elapsed. You can do this by enabling the *Enforce minimum times for fields* option within Collect's *Admin Settings* (from the main Collect menu, press your device's menu button, then choose *Admin Settings*).

While you can combine the first three options above (to count, list, and audit violations), using the last option to enforce speed limits will effectively preclude you from using the other options: if speed limits are enforced, then you can't have any violations to count, list, or audit. Thus, in most cases it may be more effective to allow violations – but then to monitor them carefully.

In combination with random audio audits and other quality checks, speed limits can be an additional tool to help assure a high level of data quality.

Designing workflow

With SurveyCTO, much of the "workflow" of filling out survey forms depends on [the design of your forms](#): users move through the various form fields that you have defined, filling them out, being subjected to constraints that you have configured, etc. But some key elements of the workflow are controlled not by individual survey forms but instead by how the SurveyCTO Collect Android app is configured on a given device. This help topic discusses that configuration.

You can find SurveyCTO Collect's settings in two places: from the Collect main menu, press your device's Menu button (or the three vertical dots in the corner on some newer devices) and choose either *Admin Settings* or *General Settings*. These settings control aspects of the user experience before, during, and after the actual filling out of survey forms.

Before

When a user runs SurveyCTO Collect, you can control the options they see on the main menu and the settings that they can edit. Once you have configured devices with the desired survey forms, for example, you might limit the main menu choices to only Fill Blank Form, Edit Saved Form, and Send Finalized Form; you might also prevent users from changing any of the settings (without a special password). See [this help topic](#) for details.

In the standard workflow, users run the SurveyCTO Collect app and choose Fill Blank Form to start filling out a form. However, you can also place a shortcut to that form directly on the device's home screen, so that users can boot up their devices and start filling out a new form with just one click. The process differs a bit by device: on some devices, you hold your finger down on an empty part of the home screen, then choose Add Shortcut, "SurveyCTO Form", and the form you'd like the shortcut to launch; on others, you click into the Apps view, then choose Widgets, "SurveyCTO Form", and, after dropping the widget onto an empty location, the form to launch.

You can also organize the filling out and editing of forms around specific "cases," where cases might be households, spouses, health centers, schools, etc. See [the help topic on case management below](#) for details.

During

While the user is filling out a form, most of the experience is controlled by the design of the form itself. This primarily involves a sequential process by which the user fills out form fields, one by one. However, [this help topic](#) discusses more advanced navigation options open to the form designer.

Typically, the user navigates forward and backward by "swiping" with his or her finger. However, there is a *Navigation* setting in *General Settings* that allows you to enable on-screen forward and back buttons; these buttons can be used in addition to or instead of swiping.

It is also typical to enforce [field constraints](#) field-by-field, as the user progresses through each survey form: if a user enters an invalid value, he or she is not allowed to continue until the entry is corrected. The *Constraint processing behavior* setting in *General Settings* allows you to override this behavior and instead defer all field validation to the point at which the survey is finalized (at the very end). For the vast majority of cases, the default – validating field-by-field – is the way to go, but it's up to you.

While within-form navigation is mostly controlled by the form designer, there is an option for the user to jump from one place to another, outside the standard sequence. This option is typically used, for example, when the user realizes that an earlier response was mistaken and wishes to jump back to fix it. He or she clicks the device's menu button, chooses *Go To Prompt* (or clicks the button with the arrow pointing to the dot), scrolls to the desired question, and clicks on it to jump there. If you want to prevent users from being able to jump around in this way, go into *Admin Settings*, scroll down to the "User can access form entry items" section, and un-check *Go To Prompt*.

In the standard, field-by-field sequence, the user cannot proceed without entering a value for any field that is marked as *required* or correcting any error if the current entry violates a [constraint](#). However, there may be circumstances where the user cannot yet fill out a particular response and wants to move on and circle back to fix it later (e.g., the response concerns entering a child's weight but the user is awaiting the necessary scale, or the particular respondent necessary for a question is currently busy). In such a circumstance, the user can press the device's menu button and then select *Skip to Next*; this is effectively a shortcut for choosing *Go To Prompt*, scrolling down to the next question, and clicking on it to jump. If you want to prevent users from being able to jump in this way, go into *Admin Settings*, scroll down to the "User can access form entry items" section, and un-check *Skip to Next*.

If an interview is interrupted for some reason, the user needs to be able to save, exit, and resume later. He or she does this by clicking the device's built-in back button to exit the form. A pop-up menu will ask whether to *Save Changes* or *Ignore Changes*. The latter option throws data away, so it has an extra confirmation, just to be sure. As long as the user chooses to *Save Changes*, he or she can resume later by choosing *Edit Saved Form* from the main Collect menu and selecting which form to edit. If going back to edit forms later is common in your workflow, see [this help topic](#) for information on how to name forms in a way that makes them easier to find and edit later.

Validating, saving, and finalizing

Before a form can be submitted to the server, it must be "finalized"; and before a form can be finalized, it must be free of any errors. In other words, all required responses must be

entered, and all entries must satisfy their respective [field constraints](#). This is because the survey designer's specifications for which fields are required and what constraints should be enforced is what, in aggregate, ensures that data is "clean" (i.e., ready for review and analysis). Finalized forms cannot generally be edited, so finalization is the last (in-the-field) step in the data-collection process.

Some projects have individual enumerators finalize forms on-the-spot, at the conclusion of their field surveys. Others prefer enumerators to save their forms and review them later, before finalizing; they may even require a supervisor to review the form for mistakes (using *Edit Saved Form*) before it can be finalized.

When the user reaches the end of a survey form, he or she is presented with a final screen with a *Save and Exit* button. That screen also has a checkbox for whether or not to finalize the form. By default, that checkbox is checked – but you can change this default by going into *General Settings* and un-checking the *Default to finalized* option. You can also hide the checkbox altogether (preventing the user from choosing case-by-case) in the *Admin Settings* (just uncheck the *Mark form as finalized* option).

If the user is finalizing at the end of the form, then SurveyCTO will first validate the entire form and confirm that all required fields are filled in and that all constraints have been met. If there are any problems, the user will be forced to go back and correct them.

If the user is saving without finalizing (perhaps in order to review the form later), the default behavior is *not* to validate the form or force the user to make corrections. However, you can override this with the *Saving behavior* option in *Admin Settings*: if you choose "Force corrections on Save and Exit," the form will be validated and the user will be forced to make corrections before being allowed to *Save and Exit* – even when he or she is not finalizing just yet. Note, however, that there is still always the option for the user to escape in the case that an interview is interrupted and he or she cannot complete the form, by using the device's built-in back button to exit and save changes. (If you don't want to force corrections on *Save and Exit*, there are two options in the *Saving behavior* setting: "Don't check answers (quick save)" and "Don't check answers (slow save)." Of the two, the "quick save" option is generally preferred; the slow-save option is available only for backward compatibility with older versions of SurveyCTO.)

Finally, if users jump around within a survey – and/or if you have set validation to happen at the very end rather than field-by-field – they can easily lose track of what has already been fully completed and what has not. While any missing required responses or constraint violations will be flagged at the very end, users may want to jump back and start cleaning up required or invalid responses earlier in the interview process. For this reason, users can at any time click the device's menu button and then choose *Validate Form*. When they do, SurveyCTO will walk through the entire survey form, from beginning to end, validating every field; if there are any problems, the user will be sent back to correct them. If the user jumps around or skips anything, this *Validate Form* option is a quick and easy way to jump forward or backward to the first point at which responses are required. As with the other

user options, you can disable this feature if you wish: simply un-check *Validate Form* in *Admin Settings*.

After

Presuming that you're not [pulling data directly off of your devices via local wi-fi](#), at some point collected data has to be sent to your server where it will be aggregated, de-duplicated, and made available for download, export, and/or publishing. This might be right away (e.g., if all of your devices have sim cards and data connections), it might be nightly (e.g., if all of your devices can reach a wifi hotspot, such as a 3g connection shared as a hotspot by a supervisor's device), or it might be less often. Whenever it happens, there are a few basic workflows.

The most standard workflow is manual: once an Internet connection is available, the user selects *Send Finalized Form* from Collect's main menu, checks off any or all listed forms, and clicks *Send Selected*. By default, Collect will only list forms that have been finalized but not yet fully sent to the server, so the user's task is generally to check all, send, and await confirmation.

In Collect's *General Settings*, there are also two options for automatic sending of finalized forms: *Auto send with Wi-Fi* and *Auto send with network* ("network," in this case, meaning 2g or 3g cellular data network). If you check one or both of these options, Collect will automatically attempt to send all finalized forms to the server whenever the allowed connection type is available. (There are also two "auto update" settings that can be used to automatically check for and download form updates – which are particularly useful when forms include pre-loaded data that changes from time to time.)