SurveyCTO

Help - Monitoring and management - Managing for quality

Below, please find topics relating to the collection of high-quality data. (These are important! You don't want to collect poor-quality data!)

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 with good survey design.

Skipping irrelevant questions

The first opportunity may seem obvious, but avoid asking questions that are not relevant. By enabling your form to automatically skip 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 for *you* to design your skip logic in a more predictable, error-free fashion. See the help on *relevance* for more on implementing automatic skip patterns.

Constraining responses

The next opportunity is data validation. Using constraints in your survey form will prevent enumerators from entering data that is obviously incorrect, invalid, or inconsistent. While you do not 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 do not want to fully disallow some kinds of responses but still want to warn the enumerator that the responses are suspicious, you can also implement a warning or confirmation (otherwise known as a "soft" constraint). Simply add a note or yes-no confirmation that is only relevant (i.e., that only appears to the enumerator) if an entry 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 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 is 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.

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 extremely 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 a random proportion of the households or facilities surveyed, confirm that they were visited by enumerators, and re-ask certain questions to compare with the original responses. 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. Use the "Monitor form data" action in the *Form submissions and dataset data* section of the Monitor tab to jump into SurveyCTO's built-in *Data Explorer*. There, you can configure and save a monitoring workbook, review aggregate data as well as individual submissions, and catch potential data-quality issues right away.

To insist that every incoming submission be reviewed (and potentially corrected) before being released for publishing or export, you can enable the review and correction workflow for any or all of your forms. Doing so builds in explicit review of all incoming data, before it is passed on to any outside systems.

For reviewing your data outside SurveyCTO, you can export all of your data or 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 the data in a more visual way. If reviewing 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 Sheets.

SurveyCTO can also help you to catch potential problems that may not be obvious in your more manual reviews. If you configure automated quality checks, SurveyCTO can help automatically detect potential problems and provide specific warnings about quality concerns. See the help topic on quality checks for more details.

Finally, you can automatically execute other automated processes (e.g., a Stata .do file to process and review the new data) whenever you export new data using SurveyCTO Sync.

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 quality assurance (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 meta-data about survey administration. This includes details on how much time the surveyor spent on each question in the survey form and the sequence with which he or she proceeded through the survey. For each audited survey, a .csv file is saved that contains this information. Once the survey data is 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

If you monitor incoming data with the *Data Explorer*, you can easily download text-audit data for any submission you view – or even click the hourglass button to see the timing information overlaid on the questions and responses.

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, audio audits will be included in separate audio files; 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 to 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 proportion 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.

If you monitor incoming data with the *Data Explorer*, you can easily download and play audio audits for any submission you view.

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 help topics for those field types (*text audit* and *audio audit*) for more details.

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 ID for the corresponding survey form (also exported in the *KEY* column).

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.mp4", where UUID is the unique ID and FIELDNAME is the name of the field at which the recording began. For other recordings, the filename will be "AA_UUID_AFTER_#S.mp4" where UUID is the unique ID 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": you can use the *minimum_seconds* column in your survey form definition to 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 violates 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 too 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 (the *x*) and the length of the audit in seconds (the *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. (And 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.)

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

Using quality checks to monitor the quality of incoming data

Even assuming that you use field constraints to prevent completely-invalid data from being collected in the first place, you will still want to monitor the overall quality of your incoming data and respond to any potential issues that arise. The quicker you identify and respond to issues, the more easily you can solve them and the higher-quality your data will be in the end. For this reason, we've made it easy to monitor incoming data using both SurveyCTO's built-in *Data Explorer* and automated quality checks. This help topic will focus on automated quality checks; see *Using the Data Explorer to monitor incoming data* for more on reviewing incoming data in the *Data Explorer*.

In just a few quick minutes, you can configure quality checks for any of your forms. Just go to the *Automated quality checks* section of your server console's Monitor tab, click the *Checks* button for the appropriate form, and click *Create quality check* to get started.

Automated quality checks can be configured to warn you about different kinds of issues:

1. **Individual field values that are too low or too high.** For example, perhaps your form technically allows respondents to be up to 120 years old – but you really don't expect respondents to be over 100. You could create a quality check that warns about any cases where an age is above 100, so that your team can follow up and confirm that it wasn't a mistake.

2. **Individual field values that are outliers.** Rather than setting a specific threshold for what is too high or too low, you can ask SurveyCTO to use statistics to determine when field values are unusually high or unusually low. You define a simple statistical threshold for what counts as "unusual," and then you get warned about submissions with values that are unusual in that way (i.e., that are outliers).

3. **Individual field values that are too frequent or too infrequent.** Looking at the full dataset, as it comes in, you might want to monitor instead the *frequency* of certain response values. For example, you might not want a gender field to contain "female" less than 30% of the time, or a "don't know" response to appear more than 10% of the time.

4. **Field means that are too low or too high.** Instead of looking at individual submissions, you might want to consider the overall mean or average of a field and warn if it is above or below a certain threshold. For example, if the average respondent income reported is above or below what you expect, there might be some problem with how it's being measured.
5. **Mean values that differ from one sub-group to another.** Instead of looking at the overall mean for a field, you might want to consider how that mean differs across sub-groups. For example, you might want to look for interviewer effects by checking to make sure that average income doesn't differ significantly depending on the interviewer.
6. **Response distributions that differ from one sub-group to another.** If you're working with discrete or categorical data rather than continuous numeric data, then you can consider the full distribution of responses in a field rather than just a single mean. Like with means, you can check to see if the distribution of responses differs across sub-groups. For example, you can see whether there are enumerator effects in the reported occupation of respondents.

SurveyCTO will report warnings to you whenever submission values, frequencies, means, or distributions in your data cause configured quality checks to fail. Read on to learn specific details about each type of quality check available, and about the quality-check reports that are generated for your review.

Types of quality check

Value is too low

What it checks: the numeric field or fields that you select, for every submission in your form data.

What it checks for: any response values that are below a numeric threshold that you specify.

Warnings it issues: value-too-low warnings for individual fields in individual submissions.

Example: check to see if the "income" field has a response value less than 1000.

Options: can specify a list of special response values (like -777 or -888) to exclude them from triggering warnings. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Value is too high

What it checks: the numeric field or fields that you select, for every submission in your form data.

What it checks for: any response values that are above a numeric threshold that you specify.

Warnings it issues: value-too-high warnings for individual fields in individual submissions.

Example: check to see if the "age" field has a response value greater than 100.

Options: can specify a list of special response values (like 999 or 9999) to exclude them from triggering warnings. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Value is an outlier

What it checks: the numeric field or fields that you select, for every submission in your form data.

What it checks for: any response values that are outliers because they are more than *x* times outside the interquartile range (IQR), for the value of *x* that you specify (1.5 is common, to identify values that are 1.5*IQR below the first quartile or 1.5*IQR above the third quartile).

Warnings it issues: value-is-outlier warnings for individual fields in individual submissions.

Example: check to see if the "income" field has a response value that is more than 1.5 times outside the interquartile range.

Options: can specify a list of special response values (like -777 or -888) to exclude them from triggering warnings. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Value is too frequent

What it checks: the overall frequency of specific field values, for the field or fields that you select, across all submissions in your form data.

What it checks for: any response values that are more frequent than a percentage frequency that you specify.

Warnings it issues: value-too-frequent warnings for individual response values within individual fields.

Example: check to see if the "gender" field has a response value of "male" more than 70% of the time.

Options: can specify whether the warnings should be classified as "critical" in quality-check reports.

Value is too infrequent

What it checks: the overall frequency of specific field values, for the field or fields that you select, across all submissions in your form data.

What it checks for: any response values that are less frequent than a percentage frequency that you specify.

Warnings it issues: value-too-infrequent warnings for individual response values within individual fields.

Example: check to see if the "gender" field has a response value of "female" less than 30% of the time.

Options: can specify whether the warnings should be classified as "critical" in quality-check reports.

Mean is too low

What it checks: the overall mean of the numeric field or fields that you select, considering all submissions in your form data.

What it checks for: an overall mean that is below a numeric threshold that you specify.

Warnings it issues: mean-too-low warnings for individual fields.

Example: check to see if the "income" field has a mean below 20000.

Options: can specify a list of special response values (like -777 or -888) to exclude when calculating field means. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Mean is too high

What it checks: the overall mean of the numeric field or fields that you select, considering all submissions in your form data.

What it checks for: an overall mean that is above a numeric threshold that you specify.

Warnings it issues: mean-too-high warnings for individual fields.

Example: check to see if the "income" field has a mean above 100000.

Options: can specify a list of special response values (like -777 or -888) to exclude when calculating field means. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Group mean is different

What it checks: sub-group means for the numeric field or fields that you select, considering all submissions in your form data.

What it checks for: any sub-groups with means that differ significantly from the means of other sub-groups, based on an ANOVA test and a significance threshold (a p-value) that you specify (usually 0.05).

Warnings it issues: group-mean-different warnings for individual sub-groups and individual fields.

Example: check to see if the mean of the "income" field differs by the sub-group defined by the "enumerator_id" field (i.e., look for enumerator differences in reported income, using the "enumerator_id" field to identify different enumerators).

Options: can specify a list of special response values (like -777 or -888) to exclude when calculating sub-group means. Can specify whether the warnings should be classified as "critical" in quality-check reports.

Group distribution is different

What it checks: sub-group distributions for the discrete/categorical field or fields that you select, considering all submissions in your form data.

What it checks for: any sub-groups with distributions that differ significantly from the distributions of other sub-groups, based on a chi-squared test and a significance threshold (a p-value) that you specify (usually 0.05).

Warnings it issues: group-distribution-different warnings for individual sub-groups and individual fields.

Example: check to see if the distribution of the "occupation" field differs by the sub-group defined by the "enumerator_id" field (i.e., look for enumerator differences in reported occupation, using the "enumerator_id" field to identify different enumerators).

Options: can specify whether the warnings should be classified as "critical" in quality-check reports.

Quality check options

You can configure as many quality checks as you wish, and each of them can cover as many fields as you like.

For each form within the *Automated quality checks* section, you can also click *Options* to configure some overall settings that apply for all quality checks configured for that form. This includes an option to *Run all checks nightly (uncheck to pause)* and an option to send email summaries of all quality-check results to a list of email addresses. If the review and correction workflow is enabled for the form, it also includes options to choose which types of submissions (approved, rejected, etc.) to include when running quality checks.

Unless you have the option to run nightly turned off, all configured quality checks will run automatically once per night. They'll also run whenever you click the *Run now* button to manually run them.

Quality check reports

If any of your quality checks are triggered (i.e., if some aspect of the form's data "fails"), then SurveyCTO will issue data-quality warnings in a report. Every time the checks are run, in fact, any new warnings are appended to the previous report – so warnings will accumulate over time, but duplicate copies of existing warnings will not be added.

You can download the quality-check report by clicking *Report* in the *Automated quality checks* section, or by clicking the link to the full report included in any email notifications you receive (if you have configured report summaries to be emailed to you).

The report itself is a .csv file that you can open in Microsoft Excel or Google Sheets. Its most important columns are:

1. *warning*: the human-readable warning resulting from each failed quality check, including which check failed and why.
2. *last-reported*: the date and time when the warning was most recently issued. Since it's common for the same data to generate the same warnings each time your quality checks are run, existing warning rows, when present, are simply updated with a new last-reported date and time (rather than appending duplicate warning rows with each run). You'll know these "current warnings" because their *last-reported* date and time will coincide with the last time a full report was generated; "old warnings," on the other hand, will have older *last-reported* values, perhaps because the offending data was corrected, the quality-check configuration changed, or the underlying statistical distributions changed such that the warnings no longer trigger.

Additional columns in the report include:

1. *critical*: 1 if you'd configured the relevant quality check to issue critical warnings; otherwise 0.
2. *dataset-id*: the unique ID of the internal dataset that was checked, which will include your unique form ID.
3. *id*: a more machine-readable unique ID for the human-readable warning in the *warning* column.
4. *warning-id*, *field*, *group-field*, *value*: details relating to which quality check you configured and with what parameters. In addition to the human-readable warning text in the *warning* column, these can help you to distinguish warnings from different configured quality checks. You might, for example, find it helpful to sort by some of these columns when reviewing long lists of warnings.
5. *row-id*, *group-id*: the unique row and/or group ID that caused the warning. When warnings are issued for particular rows or for particular groups, these columns indicate the unique ID's that identify those rows or groups.

Note that if you use the "Monitor form data" action to enter the *Data Explorer*, results from your quality-check reports will be summarized along with your data. See *Using the Data Explorer to monitor incoming data* for more.

Quality check reports as server datasets

Quality check reports are actually server datasets (see *Advanced publishing with datasets*). This means that, in addition to manually downloading them as described above, you can export them using *SurveyCTO Sync*, publish them to the cloud, merge them into Excel workbooks, or even attach them as pre-loaded data for one or more survey forms. We understand that back-office operations vary widely, so we try to keep things as flexible as possible.

Each quality check report will publish to a dataset with an ID like "formid.sampleid_qc" – but with your form's unique ID instead of the "sampleid". The first time a set of quality checks is run for a form, this report dataset will be automatically created. If you do publish a report dataset to the cloud, use the *id* column as the unique ID; that way, when some of the same warnings are issued each time the data checks are run, you will end up with one row per warning rather than multiple rows.

Limitation

When configuring quality checks for your forms, there is one key limitation to keep in mind: *you can't configure checks for encrypted fields*.

If you have encrypted your form data with your own encryption keys, you can configure quality checks for only those form fields that were explicitly marked as *publishable* (i.e., fields for which you indicated "yes" in the *publishable* column of your *survey* worksheet). This is because SurveyCTO simply can't read encrypted data.

You should take care not to mark sensitive, highly-confidential fields as *publishable*, as *publishable* fields will not be as strongly protected as other fields in encrypted forms. (They will still be encrypted in transit, but they will be readable by SurveyCTO.)

Advanced data correction workflows

The simplest approach to reviewing and correcting data within SurveyCTO is to enable the review and correction workflow for a form, choose to hold some or all incoming submissions for review, and then export or publish those submissions for downstream reporting and analysis only once they have been approved. For basic workflows, data is never corrected (it never changes) after being approved and released to downstream users. See the full help topic on the review and correction workflow for more details.

However, you can enable an advanced "un-approve" option when configuring the review and correction workflow for a form (click the *Review workflow* button for any form in the *Form submissions and dataset data* section of the Monitor tab, then enable the *Allow un-approve and un-reject* toggle). If you do so, data that has already been released to downstream users can be un-approved, corrected further, and then re-approved. In such cases, special care must be taken to ensure that downstream users handle the updated data properly.

There are a few reasons why you might want to allow reconsideration of already-approved data. One is that you might discover problems with a submission that hadn't initially been held for review; because such a submission would have been auto-approved on receipt, un-approving would be necessary in order to make any corrections. You might even configure the review and correction workflow not to hold anything for review (i.e., auto-approve everything on receipt) and rely on downstream review processes to flag submissions in need of further scrutiny and correction. Finally, you might manually approve a submission, but then discover additional problems that you wish to resolve. All such workflows are possible, if you enable un-approval.

Accommodating data changes in downstream systems

If an approved submission is un-approved, then it will no longer appear in subsequent exports from the *Export* tab, or in subsequent API requests. However, if the submission had published into a server dataset, to the cloud, into a Stata dataset, into Microsoft Word or Excel, or into any other downstream system, the un-approved data will remain in those systems. In other words, there is no "claw-back" for published data: once data has been approved and published into downstream systems, SurveyCTO has no way to retract it. This applies also to exports from *SurveyCTO Sync*: unless you explicitly select the option to clear your local cache, previously-approved data will continue exporting, even after it has been un-approved.

So if a submission is un-approved and then rejected, downstream systems will generally retain a copy of the originally-approved version of the submission. In order to clear approved-then-later-rejected submissions from your systems, you'll need to export rejected submissions and manually remove them from your systems.

If you un-approve a submission, make changes to it, and then re-approve it, it will again be released for export and publishing. What this means is that the (revised) submission will begin appearing again in subsequent data exports and API requests, and it will be re-published into server datasets, to the cloud, into Stata datasets, into Microsoft Word or Excel, etc. Some particular considerations to be aware of:

- **Re-publishing to the cloud**. If publishing to Google, be sure to configure a unique ID field (usually *KEY*) so that re-exported data merges into the same rows as previously-exported data; that way, revised data will update prior data rather than coming in as new (duplicate) rows. If using Zapier or webhooks, be sure to configure receiving systems to update existing records or rows, whenever possible, using the unique *KEY* column. (See this help topic for more...)
- **Re-publishing to server datasets**. Be sure to configure a unique ID field (usually *KEY*) so that re-exported data merges into the same rows as previously-exported data; that way, revised data will update prior data rather than coming in as new (duplicate) rows. (See this help topic for more...)
- **Re-publishing to Stata**. Edit your .do file template to change the *overwrite_old_data* local macro at the top of the template from 0 to 1. This will overwrite previously-imported data with the latest version included in the export, every time you run the .do file. (See this help topic for more...)
- **Re-publishing to Microsoft Word**. *SurveyCTO Sync* re-runs the mail merge process to re-generate Microsoft Word output with each new export, so revised data should always be included in the latest output. (See this help topic for more...)
- **Re-publishing to Microsoft Excel**. *SurveyCTO Sync* automatically merges on the *KEY* column when exporting to Microsoft Excel, so new data should automatically merge into workbooks, overwriting old data. (See this help topic for more...)

An alternative to allowing the un-approve option is to carefully review and correct each submission before releasing to downstream systems, then make any later changes directly in those downstream systems (instead of in SurveyCTO).

Advanced use of Data Explorer workbooks

In both the *Form submissions and dataset data* section of the Monitor tab and the *Your data* section of the Export tab, there is an *Advanced mode* button that allows you to enable a set of more powerful *Data Explorer* tools. Meant for more expert users, advanced mode allows you to:

- **Configure multiple workbooks.** By default, each form has one workbook on the Monitor tab and one on the Export tab. In advanced mode, you can configure as many additional workbooks as you like, tailoring each to a particular view, team, or workflow.
- **Attach datasets to workbooks.** Advanced mode also allows you to attach server datasets to workbooks, so that you can supplement incoming form data with earlier listing data, QC results from outside systems, and more.
- **Download and upload workbook definitions.** Finally, advanced mode includes *Download* and *Upload* buttons that allow you to export and import workbook definitions. These definitions are Excel spreadsheets, similar to form definitions and edited in a similar way; instead of defining fields in a form, however, these workbook definitions define summaries in a *Data Explorer* workbook.

Configuring multiple workbooks

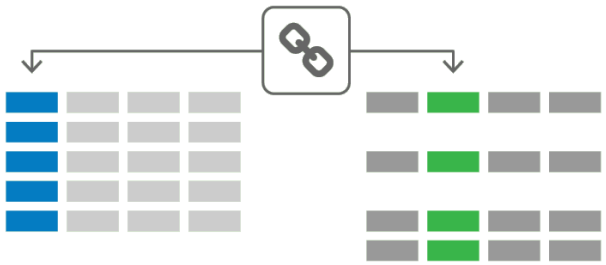
Once you enable advanced mode for a particular form, you will be able to add, edit, and delete an arbitrary number of workbooks for that form. These workbooks will all share the same form data, but you can configure field and relationship summaries – as well as filters and exclusions – independently for each workbook. That way, you can tailor each workbook to the appropriate view, team, or workflow. The only real

restrictions are that each workbook must have a title and unique ID, and you must retain the original "default" workbook. The default workbook will be used, for example, if somebody using the review and correction workflow clicks a shortcut to open the *Data Explorer* with all submissions awaiting review.

Attaching data to workbooks

When using the *Data Explorer* to review and explore incoming form data, it's sometimes helpful to merge in additional data from other forms or outside systems. For example, you might have data from an earlier listing survey, publicly-available demographic information for geographical areas, or the results of statistical checks done using your chosen analytics software. In advanced mode, you can link that data to your *Data Explorer* workbooks, so that it's available for generating field and relationship summaries.

To attach data to any *Data Explorer* workbook, first load it into a server dataset. Then, once you've enabled advanced mode for the relevant form, click the *Attach* button to attach that dataset to the appropriate workbook. When you do, you'll just need to specify a few options in order to configure merging and save the attachment.



All SurveyCTO needs to know is which form field links to which dataset column. For example, you might have a form field named *districtid* that contains a unique district ID, and a matching dataset column named *dist*. To be able to merge data successfully, the values in your chosen form field should match with values in your chosen dataset column, and your chosen dataset column should uniquely identify rows in your dataset. In the example, the district ID values in the form's *districtid* column should match the values in the dataset's *dist* column, and the dataset should only include one row per *dist*.

Once you've attached data to your workbook, SurveyCTO will automatically merge that data in whenever it can. So continuing with the same example, say that your districts dataset is named "districtdata" and that, in addition to the *dist* column, it has a *population* column. Whenever there's a form submission with a value in the *districtid* column that matches the *dist* value in a dataset row, the value from that row's *population* column will be merged in as *districtdata.population*. In effect, it'll be like the incoming form data also included the district population, in a field named *districtdata.population*.

In fact, all attached dataset columns will be merged into the *Data Explorer* workbook, as if they were fields in the form itself. When you add field or relationship summaries, the dataset fields will be added to the bottom, underneath the form fields, as additional options. And when you view an individual submission's details, the merged-in dataset data will also appear down at the bottom (if a match was found in the dataset data). The merged-in data essentially extends the form data, providing additional information that might be helpful in exploring or reviewing incoming data.

That's all there is to it. Just two minor restrictions worth mentioning: you can't merge using a form field that's inside a *repeat group*, and you can't merge using date or time fields.

Working with workbook definitions

Once you've enabled advanced mode for a form, you'll find *Download* and *Upload* buttons available to export and import workbook definitions from the Monitor and Export tabs. These definitions are Excel spreadsheets, similar to form definitions and edited in a similar way; instead of defining fields in a form, however, these workbook definitions define summaries in a *Data Explorer* workbook.

We don't expect you to manually create workbook definitions the way you might be used to creating form definitions, however. Rather, we expect you to create new workbooks in your web browser, using the *Data Explorer*. Then, once you've created a workbook that you like, you can use the *Download* option to export it

to a spreadsheet.

Once you've downloaded a workbook definition, you can archive it, share it with colleagues, or make copies. You can also edit it, of course. One common use case is to create a workbook for one version of a survey (perhaps for round one or country one), export it, tweak it a little bit (perhaps for round two or country two), and then import the tweaked version for another form or even on another server.

Each workbook includes the following worksheets:

- **summaries**: This worksheet defines all of the field and relationship summaries in your *Data Explorer* workbook, as well as how they are grouped. Each row represents either a summary or the beginning or end of a group. *Tip for copying:* be sure to correct the field names in the *field* and *field_2* columns to match with your target form.
- **global_filters**: This worksheet defines global filters, which are used to focus or narrow the summaries shown in your *Data Explorer* workbook. For example, you might filter for "F" values in your *gender* field, in order to focus all summaries on results for female respondents.
- **global_exclusions**: This worksheet defines global exclusions, which are used to omit particular submissions from the summaries shown in your *Data Explorer* workbook. For example, if you decide that a particular submission is invalid or an outlier, you might choose to exclude it from the workbook overall.
- **settings**: This worksheet defines overall settings for your *Data Explorer* workbook, including the title and unique ID. *Tip for copying:* be sure to update the title and unique ID whenever you make copies of a workbook.
- **summaries-help**, **global_filters-help**, **global_exclusions-help**, and **settings-help**: These worksheets have some quick-reference help on the rows and columns that make up the *summaries*, *global_filters*, *global_exclusions* and *settings* worksheets.

The best way to learn about the format of *Data Explorer* workbook definitions is to export a workbook and take a look, consulting the four help worksheets as needed.