

Help - Designing forms - Additional topics

Below, please find additional topics and techniques for designing your SurveyCTO forms. If you haven't already, be sure to check out the core concepts section.

Providing default entries or selections for fields

Unless you specify otherwise, all fields in your form will start out blank (or, in the case of multiple-choice questions, un-selected) until a user enters or selects something. If you would like to specify a default entry or selection for a field, you have two options:

1. **Enter a default value in the *default* column.** In this case, the exact number or string you enter will be used as the field's default entry or selection each time the form is filled out. Use this option when you have a single, fixed default that you would like to use.
2. **Enter a default expression in the *calculation* column.** In this case, the expression you enter will be evaluated the first time the field is displayed to the user, and the result will be placed into the field as its default entry or selection; the user can then accept or update this value. Use this option when you want to default a field in some dynamic way (e.g., with a value pulled from pre-loaded data).

In the form designer, you can add either kind of default whenever you are adding or editing a field: just select the type of default you would like to add, and then enter or select the relevant details.

Whenever your default is calculated based on some expression or prior field, you should configure the field to not be *relevant* until everything on which its calculation depends is ready (see *Implementing skip patterns with "relevance"* for more on *relevance*). For example, say that you have a group of fields that you would like to pre-load based on an entered household ID. In that case, you would want to put those questions within a group that only becomes *relevant* once the household ID has been entered. Otherwise, a user beginning to fill out the survey could click on the *Go To* option before entering a household ID, which would list all relevant questions – including the ones that were meant to pre-load – and because those fields' *calculation* expressions only evaluate once (the first time they are shown to the user), they would evaluate too soon (before the household ID has been entered). If, however, the fields are not *relevant* until after the household ID has been entered, this problem will not arise.

If editing spreadsheet form definitions directly, the easiest way to build a calculation expression is to use the calculation-builder available as a tool, both on the server's Design tab and in *SurveyCTO Sync*.

For details on hand-creating or hand-editing calculation expressions, see *Using expressions in your forms*.

Repeating fields (e.g., for household rosters)

For a household roster, you often want to collect answers to the same questions for each member of the household. Similarly, in a consumption module, you might want to ask certain questions about each item consumed. For a facility inspection, you might want to ask a certain set of questions about a series of rooms. It turns out that there are many such cases, where you want to repeat the same set of questions x times, where x can be anywhere from 0 to many.

You have three basic options for how to collect this kind of repeated data in a survey form:

1. **Add lots of duplicate fields to your survey.** With this approach, if you wanted to record the name for each of up to 10 household members, for example, you would create a field for each. So, e.g., `fam_name1` would ask about the first family member's name, `fam_name2` would ask about the second member's name, and so on, up to `fam_name10`. (Tip: in order to streamline the survey for those filling it out, you could first have a field that asks for the number of family members, then each `fam_name` field could have a *relevance* column to indicate that `fam_name` question prompt should show only when appropriate. For example, the `fam_name3` field's *relevance* column might include `"${numfamily} >= 3"` if the number of family members had been recorded in a field named `"numfamily"`. That way, surveyors don't need to deal with lots of irrelevant prompts.)
2. **Use *repeat groups* to ask a group of questions repeatedly, until the user indicates that he or she is done.** This method is the quickest to implement in your survey form – simply surround a group of questions with *begin repeat* and *end repeat* rows – and it offers the greatest flexibility for those filling out your form. Users can sometimes find the experience a bit challenging, however. For example, they might accidentally add another group at the end of a roster, even when they are really finished. If they do, they can then remove that group manually (Android users press-and-hold on a question in the group, web users use the *Options* menu), but it's easy to get confused.
3. **Use *repeat groups* to ask a group of questions repeatedly, but control the number of repetitions.** This is a kind of hybrid approach that simplifies the user experience for those filling out surveys, but requires a bit more work from you, the form designer. Here, you use *begin repeat* and *end repeat* rows like above, but then you specify either a fixed number (like `"3"`) or a prior field (like `"${numfamily}"`) in the *begin repeat* row's *repeat_count* column. The group of questions will then repeat exactly the number of times you have configured them to repeat.

The main advantage of Option 1 is its simplicity. First, you might prefer the way that non-repeated data gets exported in a simple "wide" format; with only one possible response per field per submission, all data is exported in a single .csv file, with a structure that doesn't change from export to export, and which can be easily mail-merged into Microsoft Word if you so desire. Second, if you have a separate field defined for each response, then you can very easily refer to those responses from anywhere in your form (e.g., a later field's label can be something like "What is `${fam_name3}`'s age?"). Within the same repeat-group, referencing fields is also simple (as in "What is `${fam_name}`'s age?" when the `fam_name` field is within the same repeat group), but referencing repeated fields becomes slightly more complicated from outside those fields' original repeat groups (you need to use the `indexed-repeat()` function). Of course, the costs to Option 1 are lots of careful cutting-and-pasting, a much longer form, and a hard limit on the number of different entries your form will accept.

Options 2 and 3 do have a fundamentally different data structure, because repeated fields can have more than one response per submission. But both the server and *SurveyCTO Sync* can re-format repeated data from "long" to "wide" format (see *Understanding the format of exported data* for a full discussion). Really the only thing you lose with Options 2 and 3 is the ability to mail-merge data into Microsoft Word.

The following sample forms demonstrate these different options:

1. *Rosters: Two methods for repeated questions*
2. *Rosters: A third, hybrid method for repeated questions (repeat_count)*
3. *Rosters: Choosing among earlier entries*
4. *Rosters: Collecting repeated information with multiple repeats*

Designing for easy navigation

When users fill out one of your survey forms, they generally start with the first relevant question, fill it out, and move forward, repeating this sequence question-by-question until they reach the end. That represents the most basic – and the most linear – way that users can navigate through your form. There are other possible methods of navigation, however, both within your survey form itself and within the built-in SurveyCTO interface provided for navigating forms.

Within a field or group label, for example, you can hyperlink to another field or group by simply adding a `#{fieldname}` or `#{groupname}` reference (hyperlinked references are always enabled for web forms; on Android, enable the *Hyperlink field references* option in *SurveyCTO Collect*). For example, say that you have two different modules for two different respondents (in groups named "module1" and "module2"), followed by a concluding section for the surveyor (in a group named "module3"), and you want to offer surveyors the option to easily fill out these modules in any order. Just before *module1*, between *module1* and *module2*, and between *module2* and *module3*, you can add *note* fields that reference `#{module1}`, `#{module2}`, and `#{module3}` in each of their labels. When hyperlinked field references are enabled, these group references will display hyperlinked labels that the surveyor can use to jump to any of the three modules.

You can use these kinds of hyperlinks to add quicker access to menu-style navigation wherever you think it's most appropriate. A few notes on employing this kind of form navigation:

1. When colored form navigation is enabled (which it always is for web forms; on Android, enable the *Color form navigation* option in *SurveyCTO Collect*), hyperlinks to groups will be colored red if one or more required fields within the group still require responses; otherwise, the links will be green if no required fields remain. This can help users who are jumping around within a survey to remember which groups of questions still require responses.
2. When entering label text into Excel, you can include line-breaks by pressing Alt+Enter on Windows or Control+Option+Enter on a Mac. (Note, however, that a lone line-break between two field or group references will be ignored. If you have a series of group references on separate lines without any other text, you will need to add commas at the end of each line, bullets like * at the beginning of each line, or something else between the lines; that way, the line-breaks will not be ignored.)
3. If you reference a *repeat group* within a field or note label (as in `#{repeatgroupname}`), then hyperlinked labels will be displayed for all instances currently in that repeat group.
4. In the label for a *repeat group*, you can reference one or more fields within the repeat group itself. Thus, for example, if you have a repeated set of household roster questions that include household member names, you can refer to the name field within the repeat group's label. This can help when reviewing or navigating the form.

SurveyCTO also provides a built-in user interface for navigating forms, which appears whenever somebody (a) returns to a form after having saved it earlier, or (b) clicks on the *Go to Prompt* option while filling out a form (on many Android devices, this is a button with an arrow pointing to a dot). This interface lists all

relevant fields in the form along with their current responses, and it allows the user to jump to any field (or to the end of the form). A few notes on how you can optimize this built-in navigation:

1. To visibly organize fields in the built-in interface, put those fields inside groups with "organized" in the *appearance* column of their *begin group* rows. Each organized group's fields will appear indented under the label for that group. (And if you want a group's fields to be "folded up" by default – so that the user needs to click on it to expand its list of fields – specify "organized-closed" in the *appearance* column of the *begin group* row.)
2. If you want the list of fields to include lists of repeat-group instances, disable the *Auto-collapse repeat groups* option in *SurveyCTO Collect*. By default, only the overall label for the repeat group is shown, and the user must click on that label to show the list of instances. (Whether this list is shown immediately or needs to be expanded, the fields within a repeat group can only be shown once the user clicks on which instance to display.)
3. If you want fields and groups to be colored green or red based on whether or not they have been completed, enable the *Color form navigation* option in *SurveyCTO Collect* (all non-required fields will always show as black, and coloring is always enabled for web users).

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. That will add a summary table underneath your repeated questions, which users can use to review data and jump around as needed.

Downloading printable copies of your forms

On the Design tab, you can download a printable version of any survey form by choosing *Download*, then *Printable version* from within the *Your forms and datasets* list.

The printable version is in HTML (web) format. If you would like to edit the form before printing it, you can save the .html file, open it in Microsoft Word, and *Save As* a Word (.docx) file.

You can optionally include notes in your printable version, to explain when groups or fields will appear (when they are *relevant*), what restrictions there are on user entries (what *constraints* apply), etc. In your form definition's *survey* sheet, simply enter your explanations into the *note* column. All notes in that column will be included in the printable version of your survey (and will not appear anywhere else).

You can also enter text to appear in the response area to the right of questions, by including that text in the *response_note* column of your form definition's *survey* sheet. For a text field, for example, you might put something like `|_|_|_|` if you are looking for two letters or numbers; or, for a checkbox, you might put a hollow square like `□` (this is a special HTML character: enter "□", without the quotes, into the *response_note* column); and finally, for a radio button, you might simply enter a capital O.

Collecting GPS data

There are three field types for collecting GPS data in SurveyCTO forms:

1. **geopoint** - for collecting one GPS position at a time, either explicitly (with the enumerator's cooperation) or invisibly. (Can collect multiple GPS positions in a repeat group.)
2. **geoshape** - for drawing boundaries on a map. (Always creates a closed polygon.)

3. **geotrace** - for automatically tracing a path, generally by walking or driving. (Can save as closed polygon or open polyline.)

Click any field type above to see more details about that particular field type, or see below for general guidance on collecting GPS data using SurveyCTO.

Android GPS performance tips

The first time you try to acquire a GPS location with a new device or in a new country, the device might take up to an hour to find the proper satellites. Leave the device for a while, so that it can work. Once it finds the satellites, it should become much faster from then on.

Your device may still take a little while to capture an accurate location – especially after powering down or rebooting. If you want your form's users to be able to more quickly capture accurate GPS positions, you can configure *SurveyCTO Collect* to "pre-warm" the device's GPS unit at the start of any form that captures GPS positions. This will consume some extra battery power at the beginning of the form, but it will make it more likely that GPS positions captured later in the form can be captured quickly. To pre-warm GPS, go to the Collect main menu, click your device's menu button, select *Admin Settings*, scroll down to the bottom, and select one of the *Pre-warm device location* options; you can choose to pre-warm more or less aggressively (for greater accuracy and more time, or for lesser accuracy and less time, depending on how much battery you want to spend on pre-warming).

Advanced offline mapping

The geoshape and geotrace field types involve interacting with a map, as does the geopoint field type with the "maps" or "placement-map" *appearance*. On Android, this generally requires a working connection to the Internet, so that map tiles can be downloaded on-the-fly to cover the necessary geography. It is possible to pre-load all of the necessary map tiles onto Android devices, though, and configure them to be able to use those tiles fully offline – but a warning: it's a slightly complex, technical process. You will likely need some level of technical GIS training to be able to configure and manage offline maps support.

Required: offline map tiles in raster-based MBTile files

SurveyCTO's mapping features include support for the offline presentation of layers using tiled map data. These layers can be loaded from the map interface, and their features will be overlaid onto the mapping engine's "basemap" – without the need for an Internet connection. The layers themselves can contain any geospatial data that you are already working with or are interested in presenting, whether that be high-resolution satellite imagery, population heatmaps, or wireframe drawings.

SurveyCTO Collect supports the open-source MBTile format, which provides an efficient specification for storing raster- and vector-based map tiles in a SQLite database. Please note that, at this time, *Collect only supports raster-based MBTile files*: vector-based files will unfortunately not work.

Raster-based MBTile files can be created using open-source tools such as TileMill, as well as through commercial products such as Mapbox Studio or MapTiler. The MBTile specification's website also maintains an "Implementations" page, which lists software that is known to produce valid MBTile files. (Procurement or creation of the actual MBTiles files is beyond the scope of the help available here in SurveyCTO.)

Loading raster-based MBTile files onto devices

In order to make your offline map tiles available on devices, you must manually copy or upload your MBTile files to a subdirectory within *SurveyCTO Collect*'s "layers" directory, which is itself a subdirectory of the "SurveyCTO" directory on any device that has opened *SurveyCTO Collect* at least once. For each layer you want available to users of Collect, create a subdirectory under /SurveyCTO/layers, and give it a user-friendly

name since users will choose which layer to use by choosing the directory name. So, for example, tiles saved in a directory named `"/SurveyCTO/layers/Northeastern region satellite"` would appear to users as "Northeastern region satellite".

Tips for loading: Depending on your Android device, whether you are storing the *SurveyCTO Collect* app on internal memory or an external SD card, and what file transfer method you are using to actually move the MBTiles files, the mounted path for the root SurveyCTO directory may not be immediately apparent. Applications stored on internal memory are typically mounted at `/data/app`, while applications stored on external SD cards are typically mounted at `/mnt/sdcard`. If your MBTiles are already accessible on your Android device via a file-sharing service such as Google Drive or Dropbox, we recommend that you use an Android file manager app such as *ES File Explorer* (freely available from the Google Play Store) to copy files into the right places. If you plan on transferring MBTiles over a USB connection (which may be preferable when dealing with multi-gigabyte MBTile files), we recommend following your operating-system-specific instructions, including use of the Android File Transfer program on Mac computers or the Android file transfer support built into Windows computers. If you have any trouble establishing a direct USB connection with your Android device, we recommend consulting your manufacturer's help documentation, and specifically search for how to place your device in file transfer (MTP) mode.

Configuring offline mapping options

SurveyCTO Collect has configuration options for offline mapping – but they're hidden by default. To un-hide them, go to the Collect main menu, press the device's menu button (generally three vertical dots), and select *Admin Settings*; from there, scroll down to the "User Can Access Change Settings Items" section, continue scrolling to the "Advanced mapping options" item, and place a checkmark in the box labeled "Uncheck to hide from General Settings". Then if you go back to the main Collect menu, click the menu button, and choose *General Settings*, you will be able to scroll down to find the "Advanced mapping options" section. That's where you configure the device's offline mapping options, including:

1. **Mapping engine** - for choosing between Google Maps and OpenStreetMaps presentations. Google Maps is the default selection, and we recommend that you keep this setting unchanged unless you have a compelling reason to use OpenStreetMaps or one of their specific map styles.
2. **Basemaps** - for choosing the style of the basemaps to use. Your options here will depend on the mapping engine you choose; for Google Maps: Streets (default road-map view), Satellite (Google Earth satellite view), Terrain (physical topography), or Hybrid (mixture of the streets and satellite styles); for OpenStreetMaps: OSM Streets (default road-map view), USGS National Map Topo (U.S.-focused, quadrangle-format), USGS National Map Sat (U.S.-focused, Landsat-derived), Stamen Terrain (hill shading, natural vegetation colors), CartoDB Positron (light-colored, designed for data visualization), or CartoDB Dark Matter (dark-colored, designed for data visualization).
3. **Offline Map Tiles** - to remind you that offline map tiles should be loaded into subdirectories of the `/SurveyCTO/layers` directory (discussed further above).

Using offline map layers

Once you've loaded your .mbtiles files into one or more subdirectories off of the `/SurveyCTO/layers` directory, users in a map view for a geoshape, geotrace, or geopoint field will be able to click the "stack of square sheets" button to select the appropriate map layer.

Tips for offline maps

Following are additional details and tips for succeeding with offline map tiles:

- If you were involved with the creation of your MBTile files, you might already be familiar with a particular basemap that was displayed by the creation software to help you position your tiles. The creation software's basemap was only used as a visual reference; internally, the software translated your arrangement to geospatial coordinates, which were then associated with each individual tile in your file. Because of this, you may notice that our interface's basemap does not change to the presentation you are familiar with from the MBTile creation software. This is expected behavior: MBTile files rely upon the mapping engine to supply the basemap.
- MBTile files store the zoom levels at which the layer should become visible on the basemap. This metadata specifies both the highest zoom level (when the layer first becomes visible) and the lowest zoom level (when the layer stops being visible). Depending on how the MBTile files were created, these zoom-level rules might not be intuitive to new users. For example, if an MBTile file specifies the highest zoom level too low, then a user might be positioned directly over where the layer should appear, but technically be one zoom level too far away – and the layer will not be visible. Similarly, if an MBTile file specifies the lowest zoom level too high, then a user might be positioned directly over where the layer should appear, but technically be one zoom level too close – and again, the layer will not be visible. Finally, if the zoom-level range is too narrow, the user might have difficulty finding the layer, unless they first pan to the precise area where it should appear, and slowly advance zoom levels until it comes into view. Because of these considerations, when using a map tile for the first time we recommend that you deliberately pan to the correct area first, slowly zoom in until the layer is visible, and subsequently investigate at what zoom levels this layer will be visible.
- When, in a map, you first select a layer to show, the layer might not be immediately loaded by the mapping engine. In order to force a refresh of the basemap and load the layer, zoom one level in or out on the map. After zooming in or out, the layer should become visible, and you can then return to your previous zoom level.
- Complex MBTile files with high-resolution imagery may contain several gigabytes of data, and this may contribute to significant delays in the layer appearing on first load. Additionally, to a lesser extent, there may be some delays with individual tiles loading as you pan and zoom around the layer.

Custom tiles for online use

Finally, please note that the instructions above assume that you would only want to adjust the mapping engine or load custom tiles for offline mapping. However, you can follow all of these instructions also in online settings, if you want to use OpenStreetMap maps and/or your own custom map layers. The instructions are the same either way.

Collecting respondent signatures

For documenting informed consent or similar purposes, you may want to collect respondent signatures. To do this, simply use an *image* field with "signature" specified in the *appearance* column. When users fill in forms using an Android device, they will be able to sign using their finger or a stylus; when they fill forms on the web, they will be allowed to upload a signature file.

Encrypting form data

If you are collecting sensitive, personally-identifiable data, then you will want to encrypt it so that your authorized team members are truly the only ones with access to that data. The model of encryption supported by SurveyCTO is the following:

1. As soon as a surveyor marks a filled-out form as "finalized," the form's contents will be encrypted using your public encryption key – except for those fields explicitly marked as *publishable* (those fields with a "yes" in their *publishable* column). Fields explicitly marked as publishable are left unencrypted so that they can be conveniently published to cloud services or directly downloaded from the server.
2. From that point forward, the form can no longer be edited because not even the device on which the form was filled can decrypt it (the "device" in this case can be an Android device or the SurveyCTO server, in the case of web forms).
3. Whenever form data is transmitted via a 3g or other Internet network, it is encrypted in transit using SSL. This is true for all form data regardless of whether the form itself is configured to be encrypted – so, in a sense, encrypted forms are doubly-encrypted (once with your public key, and then again with SSL).
4. The SurveyCTO server stores the form data, but it remains encrypted and therefore unreadable by the server (and by anyone who might conceivably compromise that server, legally or otherwise). (Again, though, fields explicitly marked as *publishable* will remain readable by the server.)
5. When the form data is downloaded by an end-user using *SurveyCTO Sync*, the *SurveyCTO Data Explorer*, or the *Download* option on the Export tab, it is again doubly-encrypted, with both your public key and the SSL protocol used for secure data transmission.
6. Sync, the Data Explorer, and the server download option all store local backups of the form data in local storage, but it remains encrypted and therefore unreadable.
7. The end-user who downloads the data can use your private encryption key to decrypt and read that data. For added security, this user may choose not to store that private key on an Internet-connected computer; the user may use Sync to transport the data to a secure "cold-room" computer, and only then decrypt the data, export it for further processing, and begin the analysis.

The key thing is that the SurveyCTO model of encryption relies on *you* generating the encryption keys with which data is encrypted; this assures that you and you alone are able to access your data. So the first step is to generate your own public/private encryption key pair, which you can then use to encrypt all of your sensitive forms.

To create a new key pair, navigate to the Design tab, scroll down to the *Your forms and datasets* section, click the *Tools* option at the very top of the section, and then click the *Create new key* option. Once you click to open the key generator into a new browser window, you can disconnect from the Internet if you wish to ensure that the key pair is generated and known only locally, on your computer. In the key generator, you will name the key pair, then "download" the private and then public key files ("download" being in quotes because all of this is happening, for privacy, locally in your web browser and *not* on our server). By default, the key files are named `keyname_PRIVATEDONOTSHARE.pem` and `keyname_public.pem`.

keyname_public.pem is the public key used to **encrypt** data. As the name suggests, it is public – so you don't need to worry about safeguarding it.

keyname_PRIVATEDONOTSHARE.pem is the private key used to **decrypt** data. This private key you want to guard very closely. In fact, you may want to generate the key pair on a "cold room" computer that is extraordinarily secure (e.g., disconnected from the Internet) and never have other copies of the private key in

less secure locations. (To generate a key pair on a cold-room computer, run *SurveyCTO Sync* and select *Create a public/private key pair...* from the *Tools* menu.)

While you do want to be very careful with your private key, make sure that you do not lose it. The private key is your only way to decrypt data. If you lose it, you will also lose the ability to decrypt data. Thus, you will need some very secure way to back up or otherwise safeguard your private key against loss. If you have it only on a cold-room computer, for example, and that computer fails – how will you decrypt your data?

Once you have a key pair, you are ready to configure your forms for encryption. To start a new encrypted form, just click + and then *Start new form* in the SurveyCTO server's Design tab, enable the *Advanced options*, and mark the checkbox to indicate that you want the form to be encrypted; in the next step, upload your *public* key file when prompted (*not the private key: that you never upload to anybody*). Your new form will then include, on its *settings* worksheet, a copy of your public key. That will then be used to encrypt all finalized forms.

You will design and deploy these encrypted forms just as you do non-encrypted forms, and surveyors will fill them out and submit them in the usual way. The only real difference will come in when you are ready to publish, download, or analyze your data.

Accessing your encrypted data will require use of your private key. When needed, the server console will prompt you for your private key file. When using *SurveyCTO Sync* to export your encrypted form's data to local .csv files, you will need to specify the location of your private key: use the *Browse* button to point Sync to where it resides on your local system. Without that private key, Sync cannot decrypt and export your data. (It can download the data and store it locally, but it cannot decrypt or export it.)

If you are keeping the private key only on a cold-room computer, follow these steps to download, decrypt, and export your data:

1. On an Internet-connected computer, run *SurveyCTO Sync* with *Server* as the data source and *External drive* as the data destination. When you download data, it will download and sync to an external hard drive (either a thumb drive or an external hard drive).
2. Eject the external drive, bring it to the cold-room computer, and plug it in.
3. Finally, run *SurveyCTO Sync* on the cold-room computer with *External drive* as the data source and *Local CSVs* as the data destination. Also tell Sync where to find the private key file using the *Browse* button.

If you want to publish some parts of an encrypted form to a server dataset (e.g., to feed directly into other forms) or publish some parts to the cloud (e.g., to monitor key indicators via a Google Sheets dashboard), you will need to explicitly mark some fields as *publishable* so that they will remain unencrypted (by putting a "yes" in their *publishable* column). Those fields will still be encrypted in transit, but they will be technically readable by the SurveyCTO server. (Please note, however, that file attachments – like photos or audio recordings – are always encrypted: even if they are marked as *publishable*, they cannot be accessed without the private key.)

Translating a form into multiple languages

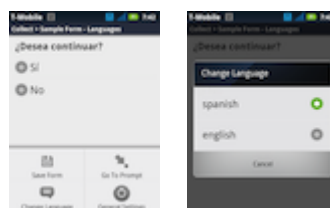
Though SurveyCTO forms default to having just a single language (English), you can translate them into any arbitrary number of languages.

In the form designer, click to edit *Form settings* to rename, add, or delete any form's languages. Once there is more than one language defined for a form: the form settings will control which language is the default for users filling out the form; a drop-down language selector will appear on the main designer screen, for the language that should be used for the summary of fields on that screen; and additional options will appear everywhere to include translations for labels, hints, images, and more.

In the spreadsheet form definition, follow these steps to add additional languages to any of your forms:

1. On the *survey* worksheet, use the *label*, *hint*, *constraint message*, *relevance message*, *media:image*, *media:audio*, and *media:video* columns for whatever you want to be the form's default language (i.e., for the language that appears whenever somebody starts filling out a new form).
2. Likewise, use the *label* and *image* columns on the *choices* worksheet for your default-language text and imagery.
3. On the *settings* worksheet, specify the name of your default language in the *default_language* column (in row 2). This tells SurveyCTO the name of this default language, so that it can show it in the user interface.
4. On the *survey* worksheet, add additional columns for each non-default language translation, and name those columns (in row 1) the same as the existing columns – but with *:language* tacked onto the end of each name (e.g., *label:language*, *hint:language*, or *constraint message:language*, where "language" is "Tamil", "Shona", "Spanish", or some other language name of your choosing). Be sure to leave one copy of each column with no language name in its name, however, so that there is a "default" translation for each column.
5. Likewise, add *label:language* and *image:language* columns to the *choices* worksheet, for whichever additional, non-default languages you would like to support. (And leave the plain *label* and *image* columns for your chosen default language.)

And that's it! Once your form includes support for multiple languages, users can switch between those languages when filling out the form. From any page of the survey – including the opening screen – they can choose *Change language* to choose among any of the supported languages. (On Android, they click their device's menu button to find the *Change language* option; on the web, it's in the *Options* menu.)



For a working example, see *Languages: Including translations*.

Importantly, Android devices have limited (and idiosyncratic) support for non-Latin fonts. For any non-Latin font, be sure to only use UNICODE fonts, and to carefully paste those UNICODE fonts into the form's Excel file or Google Spreadsheet. Finding a font supported by your devices may require some experimentation, and you may need to source devices locally if international devices do not support the script you require.

For the *SurveyCTO Collect* user interface itself on Android (buttons, menu options like *Change language*, etc.) the app attempts to respect the locale set in the Android device's overall settings. Thus, for example, if your locale is set to Español, then the Collect user interface will be in Spanish. At present, the web interface's buttons and menu options are only available in English, but users can use their browsers (and tools like Google Translate) to translate web forms.

Finally, a helpful hint: when you add columns to your worksheets, you probably want to add them all the way at the end (to the far right). Otherwise, if you later copy and paste fields from other forms, the earlier columns won't line up and the pasted fields will have information in the wrong columns (e.g., the *relevance* expression might end up in the *hint* column).

Hiding forms during testing

By default, any form that you deploy will appear to all users clicking *Get Blank Form* in *SurveyCTO Collect* or downloading data with *SurveyCTO Sync*. However, forms with titles beginning with "TEST - " will *not* appear by default; those forms will be invisible. So simply put "TEST - " at the beginning of your form titles in order to hide them during your development and testing.

To show these hidden test forms on selected devices and computers, just enable the *Show test forms* option in the preferences for *SurveyCTO Sync* and in the *Admin settings* for *SurveyCTO Collect*.

Validating forms offline

If you have a slow Internet connection or are working on your spreadsheet form definition while traveling, you may prefer to validate your form offline, without having to upload it to the SurveyCTO server.

To validate your form using *SurveyCTO Sync*, go to the *Tools* menu, scroll down to *Form tools*, and select *Validate form*. In the pop-up window, use the *Browse* button to tell Sync where to find the form definition. Sync will then validate your form and report any errors or warnings, just as if you had uploaded it to the server.

To actually use your form to collect data, however, you will still need to install it onto one or more devices. The standard approach is to upload your form to the server and then download it onto a device (as discussed in *Getting started*). If you need to install a form onto a device without an Internet connection, however, you can do so via a local wi-fi network: see *Operating more fully offline (without the Internet)* for details.

Requirements

If you are running *SurveyCTO Sync* on Windows, it may need to download an add-on in order to validate forms offline. Thus, be sure to try the offline form validation feature at least once while you are still connected to the Internet; once you succeed the first time, it will not require a connection for subsequent uses. On OSX, no add-on should be required. On Linux or other platforms, you will need a functioning Python distribution in order to validate forms offline.