SurveyCTO

Help - Designing forms - Performance and debugging

We hope that the advice below will help you in debugging and optimizing the performance of your forms.

Debugging form errors

Whenever you save your form in the online form designer – or upload your spreadsheet form definition to the server – it is converted into an internal XML format and then checked for certain types of errors. If an error check fails, you will be alerted with an error message and you will be unable to deploy your form.

Some of the upload errors may seem cryptic, so troubleshooting can take some care and patience. (We're sorry, and we're always working to improve this!) Some suggestions:

1. Scroll to the top of the error message, and look for clues about the nature of the error. In most cases, the error is associated with one of your fields, and the name of that field will be listed.
2. Use the search feature in the online form designer, in Excel, or in Google Sheets to find the source of the error. For example, if the error says that a certain field name cannot be found, search for that name to see where it is referenced, then fix that reference (it was probably just a small typo in an expression).
3. If you're editing the spreadsheet form definition directly, double-check your field types to make sure that you don't have a typo in one of them. See *Core concepts* for a list of valid field types.
4. Also double-check that there are no spaces or punctuation in any of your field names (in the *survey* sheet's *name* column), in any of your choice-list names (in the *choices* sheet's *list_name* column), or in your form ID (in the *settings* sheet's *form_id* column).
5. Most errors are caused by typos in expressions. We recommend that you create your expressions by using the online form designer, or by using *Build constraint*, *Build relevance*, and *Build calculation* tools available in the Design tab's *Your forms and datasets* section. To confirm that a particular expression is causing an error, you can delete it from your form, then re-try the upload; if it uploads without errors, then you know that you need to correct that expression.
6. To positively identify the source of an error, you can always back up your form, then iteratively trim parts of the form, re-trying the upload every time you trim something new. When an upload works, you know that the problem was in the last part trimmed.

You will be able to find and fix any error, but it may require some systematic troubleshooting. With each new SurveyCTO release, we seek to make errors a bit more clear and troubleshooting a bit less laborious. Suggestions are always welcome.

Maximum form length

There is no fixed limit for how long your survey forms can be. On Android, the primary limiting factor will be your data-collection device: if your form is too long for your device's CPU power, it may become very slow, and if your form is too long for your device's memory, it may crash.

See *Choosing an Android device* for factors to consider when choosing a data-collection device. Importantly, devices running versions of Android prior to 3.0 severely limit the amount of memory that any one app can access; from Android 3.0 on up, apps can access more memory – but the device itself still decides how much of the overall memory to allow one app to utilize.

To get an idea of how different devices perform with very long forms, you can test devices yourself using a sample form with 500 fields, a sample form with 1,500 fields, or a sample form with 3,000 fields.

Note that when you upload a very long form to the server, it may take several minutes to process. Note also that when you choose to fill out a very long form in *SurveyCTO Collect*, it may take a long time to load; once it has been loaded once, however, it will usually load substantially faster when selected again.

Generally, you can shorten a very long form by using *repeat groups*, and/or by splitting it into multiple forms linked on the back-end with a unique ID (perhaps implemented using a scanned barcode). If survey modules will always be filled out in sequence, you can even link forms automatically in your back-end processing code based upon their start-time, end-time, and device-id values.

There are other things that you can do to improve a form's performance. See *Optimizing form performance* just below for some ideas.

Optimizing form performance

If you have a very long form that is not performing well, first consider the discussion in the help topic just above this one. The solution may be to use a newer, more powerful device if your form is very demanding; or, you might choose to split one very long form into several smaller ones.

However, you might also be able to improve the performance of your form by altering some aspects of its design. To begin with, it is helpful to understand the two key reasons why users might encounter sluggishness when filling out a form:

1. **Memory:** All form fields, choices on the *choices* sheet, and translations are stored in memory while a form is being filled out. This includes repeated fields, so if you have a 100-field repeat group that's repeated 20 times, then that's 2,000 fields in memory (even if the fields are not currently relevant and so aren't visible). To protect the OS and other apps, Android only gives a limited amount of memory to each app – and when memory gets low things start getting very sluggish because of how Android tries to conserve remaining memory.
2. **Expression dependencies:** Every reference to another field or group within a relevance, calculation, or constraint expression creates a dependency that is watched for potential changes, so that the appropriate expressions can be re-evaluated whenever something on which they depend might have changed. So if Field A refers to Fields B and C and Field C refers to Field D, then moving forward or backward from Field D will trigger re-evaluations of the expressions for Fields C and then A. If you have 3,000 fields in memory (including all instances of repeated fields), then there will be up to 3,000 relevance, constraint, and calculation expressions that are always being considered for re-evaluation (including the relevance expressions for all of those fields not currently visible). So when you move

from one field to the next and you experience a delay, ask yourself: what can it be doing just now? Think about the field that you're moving away from and what might depend on that field's value. Sometimes, with a long, complex form, one field can trigger several other fields to recalculate, and those can trigger several more, and the dependencies run so deep that hundreds or even thousands of expressions will be re-evaluated. Thus, in this way, performance will depend on the logic of your form.

Understanding these causes of sluggishness, there are a few things that you can do to improve performance:

1. **Move choices to pre-loaded .csv files:** If you have a large number of choices on your *choices* worksheet (e.g., over 1,000), you should consider moving longer lists to pre-loaded .csv files. This will move the longer lists out of memory and will, in general, improve performance. See *Loading multiple-choice options from pre-loaded data* for details.
2. **Reduce the number of repeats:** Some form designs involve a large number of repeated groups, most of which are never visible. For example, say that you want to ask a long series of questions about certain assets that might be found in a household; if there are 175 different types of assets, you might create a repeat group that's repeated 175 times, with only the relevant assets visible for any given household. In principle, this is a fine design – but the 175x fields in memory can cause the form to perform poorly. Generally, in these cases, there is an alternative design that achieves the same result without so many fields in memory. (In this example case, you could have a multiple-choice question with 175 choices, then repeat the list of questions just once for each selection made by the user. That way, there are only as many fields as necessary to cover the assets in any given household.)
3. **Reduce the number of dependencies:** If you've designed your form such that long, complex expressions create such intricate dependencies between fields that any change to one field results in re-evaluation of dozens or hundreds of complex down-stream expressions, the solution may be to simplify your form. However, you should be able to take full advantage of the electronic medium to constrain user entries, catch potential errors, etc. – so in some cases the solution may be to deploy your survey on more powerful devices that can handle whatever logic your survey requires.

Neither Android nor SurveyCTO is infinitely powerful, so there will always be limits to what can perform well. But devices grow more powerful by the day, and we at SurveyCTO are always pushing the performance frontier outward. If you are using an older version of SurveyCTO, you should try upgrading to the latest version to see if that performs better. We try to include performance improvements whenever we can.

Editing performance in Excel

SurveyCTO's spreadsheet form templates and sample forms contain *conditional formatting rules* that will draw lines between lists on the *choices* tab and highlight fields on the *survey* tab. Generally, this formatting is helpful – particularly when you're starting out.

However, as you copy, paste, and insert rows in your form, Excel quietly duplicates all of the conditional-formatting rules. Everything might look fine on the surface, but behind the scenes Excel gets slower and slower as it struggles to handle a greater and greater number of duplicated formatting rules. At some point, this can become a real problem.

If your spreadsheet form definition has been edited so much and has gotten so big that Excel has become very sluggish, you might want to delete all of the conditional-formatting rules. To do so, you can click the *Conditional Formatting* toolbar button, then select *Clear Rules From Entire Sheet*. You will lose the nice

formatting, but Excel will become speedy again. (Instead, if you're an Excel expert, you can just delete all of the duplicate rules and restore to one set that covers the entire sheet.)