

















In-Depth Guide

Why use the REDCap Mobile App?

1. Collects your data *offline* with REDCap data collection instruments and adds it to a REDCap project.
2. Great for home visits, places with poor signal, or trips to the developing world.
3. Launched in April, 2015, it currently receives around 750 (and increasing) downloads per month. It's been used on projects large (100+ tablets) and small (1 phone). It's been downloaded in 100+ countries.
4. It's great for places where you can't be online but still want to collect data.
5. It's not REDCap. You can't design records for a study. That has to happen online.
6. It's a cool *tool* that extends REDCap for offline collection.
7. Do you notice a recurring theme? *Offline*. This is where you go when you do stuff *offline*.

How does REDCap and the REDCap Mobile App compare?

Features	REDCap Mobile App	REDCap
Offline Data Collection		
Online Survey <i>CATS (Computer Adaptive Test Surveys)</i> <i>Double data entry</i> <i>Survey queue</i> <i>Randomization</i> <i>Viewing non-image files</i> <i>Inline audio/video</i> <i>Stop actions</i> <i>Thank-you text</i> <i>Downloadable files in file fields</i> <i>Dynamic SQL queries</i>		
Action Tags		
Translations		
1000+ field forms		
Longitudinal projects		
GPS action tags		
Barcode scanning action tags		

Where do you start?

1. Learn the lingo.
 - a. On REDCap, there is a data collection *form*. There are also *surveys*.
 - b. On the REDCap Mobile App, there are *instruments*.
 - c. There are *REDCap users* and *app users*. They are two different entities. One REDCap user can sponsor many app users on one project, though institutional policies vary here. Similarly, one app user can have projects from different REDCap users, even at different institutions.
 - d. The *QR Code* is the black-and-white fuzzy square that you use to access your project. Some institutions only use a 10-character code that lasts for only 10 minutes. Either one will access all you need for your project.
 - e. I use the words *Project Setup* and *Project Configuration* as meaning the same thing. The word “configuration” would not fit on the button in the app, so I used the word “setup” instead.
2. Build the REDCap project.
 - a. Get an API token for your project.
 - i. Some institutions make you wait a while for this to be approved. You might as well ask early. Each REDCap user needs her/his own token for each project.
 - ii. Do check Mobile App Rights, but you don’t have to check API Import or API Export Rights.
 - iii. If you want to download your data for modification, check Download Data rights. If you just want to upload new data, leave this unchecked. The app will handle the record ids for you.
 - b. Instrument design hints
 - i. You can include images and any fields (including signatures).
 - ii. You can’t include video or audio as those will not be downloaded.
 - iii. You can’t include external PDFs as those will not be downloaded.
 - iv. Signatures *will* be downloaded as those seem like important enough information for a small amount of bytes.
 - v. Regular file fields will not be downloaded as the purposes seems to be storing them on the server. Their filenames and sizes will attempt to be downloaded.
 - vi. Don’t make your forms too long. Shorter (< 100 fields) fields seem to work better with users. 1000+ field forms can induce frustration. Making your form too long can crash your device. The largest form I’ve ever seen rendered had 8500+ fields. It was a bad idea. Put in an additional form and use the “Save and Go To Next Instrument” button.
 - vii. Heavy amounts of branching logic and calculations will slow down the rendering of your form. If you’re ok with that, the app is ok with that, too.

- c. Get a mobile device and download the app.
 - i. Devices
 1. Devices can be phones or tablets.
 2. The most popular Android devices are the Google Nexus 7/9 and the Samsung Galaxy Tablet.
 3. iOS devices currently break down about 2/3 iPhones, 1/3 iPads.
 4. Android devices handle large projects better, for some reason.
 5. iOS devices receive the best marks for hardware, but they are also the most expensive.
 6. Android devices have more of a global appeal because Google is in more countries than Apple.
 7. WiFi-only devices are ok in places where WiFi is readily accessible. Devices that support data transmission over cellular networks are required for places without much WiFi.
 - ii. Places to download the app. Search for REDCap Mobile App at:
 1. Google Play Store for Androids.
 - a. Auto-upgrade possible
 - b. Latest version only
 - c. Can push multiple versions a day.
 2. Apple App Store for iOS.
 - a. Auto-upgrade possible
 - b. Latest version only
 - c. Can push two-or-so versions in one week.
 3. (Soon) Vanderbilt REDCap Store for Android and Kindle Fire.
 - a. Manual upgrade
 - b. Select and install your version
 - c. Prior versions available; can set institutional recommended version
 - d. Awaiting new EULA from Vanderbilt's Tech Transfer team.
3. Open the app and Set Up a Mobile Project
 - a. QR Code
 - i. Take a picture of the QR code. This transmits information from REDCap to the app.
 - ii. Then, if you are successful, you can download the project's configuration and, if desired, some data.
 - b. Alternate Method
 - i. This involves typing in a 10-character sequence.
 - ii. Half of this sequence is stored on your server; half at Vanderbilt.
 - iii. Requires access to redcap.vanderbilt.edu over the current network.

- iv. The characters are valid for only 10 minutes.
 - v. It transmits the same information (encrypted) as the QR code.
 - vi. Similarly, it sets up a project's configuration.
4. Common Setup Problems
- a. *Could not get form rights (B)!* This means that it could not access your server for some reason (often a firewall). If there is no firewall, retrying the process will often work.
 - b. *Timing out on downloads.* There are about 5-10 different download queries that happen to your server. If one of these fails, the app will fail to proceed. Often, there is no error message. It just dies. Try finding a spot with better Internet access.
 - c. *Does not recognize QR code.* Sometimes, the camera of your device will stubbornly not recognize the QR code. Here, you have to revert to the Alternate Method.
5. Downloading Data (optional)
- a. There are many ways to download your data. If the dataset is small, you might want to download *all* your data.
 - b. If, however, you have over 1000 records or if you have a large number of fields within each record, you might want to download a partial set of data.
 - i. You can check which records you want and leave the other records on the server.
 - ii. Some people turn auto-numbering off and preface their records by a group name (like S1-1234 vs. D1-5678). Then they search for all the records with S1- in their name. They check all of these, and they download all of these.
 - c. Others still download by a particular form-status. That is, they download only records with one form/instrument filled out and its status is marked as "Incomplete," "Unverified," or "Complete."
 - d. If your data download is interrupted and you return to the download process with the same project, you should be allowed to restart your download near the point you left off. This is meant to be helpful in Internet-poor places.
 - e. The download data process proceeds in batches so as to pick up where you left off. Retries are built into the routine as well. That said, the download algorithm cannot compensate for terribly poor signal.
 - f. We are trying to build in a system that works with cellular and WiFi maps from all over the world. Hopefully, this will tell users where to go if they are having troubles.
6. Project Organization
- a. Projects are organized into instruments and into events if they are longitudinal.
 - b. All instruments within an event (longitudinal) or a project (classical) are in a sequence. Save and Go To Next Instrument moves to the next instrument.
 - c. Securing an Instrument

- i. There is a way to secure the app and hand it over to a participant. On the instrument-rendering page, there is a button to Secure the Instrument.
 - ii. This will lock down the app so that only the instrument is available. The device is still available for access, but the app requires a password (at the bottom) to resume to normal mode.
 - iii. This will hopefully allow users more privacy.
 - d. Securing an Event or a Project
 - i. This is the same idea as securing an instrument, only the Save and Go To Next Instrument button is present when there is another instrument in the event/project.
 - ii. This allows a series of instruments to be filled out by a participant in privacy.
 - e. File fields become capture fields for pictures, audio, and video. There is a 30 MB limit on the file size, which corresponds to the recommended file upload limit in REDCap.
 - f. To disable the “Create New Record” button, disable the sponsoring REDCap user’s Create Record rights in REDCap. A similar process is in place for Rename Record rights.
 - g. Records and fields are marked either new, modified, or unmodified. This becomes relevant at upload time.
 - 7. Sending Data (otherwise known as Uploading or Syncing Data)
 - a. This has by far the most complex code in the app. As such, it is the part of code most prone to bugs. If you detect a bug in this process, do the following:
 - i. Logout.
 - ii. Login with your normal PIN, only add a ‘00’ to the end. This will make it an 8-digit PIN.
 - iii. This throws the app into Testing Mode. There is a button to submit a diagnostic report at the bottom of each screen.
 - iv. Reproduce your bug in Send Data if you can.
 - v. Press the button at the bottom of the screen.
 - vi. Send a Bug Report via the main menu of the app or by emailing redcapapp@vanderbilt.edu. We need a normal-language story of what occurred around the bug in order to make the diagnostic reports make sense.
 - b. Records are not considered “real data” until they are synced into REDCap. They are just a part of the app.
 - c. The algorithm for this process is to send New records first and then adjudicate Modified records.
 - d. New records
 - i. For projects using autonumbering, a list of record IDs is downloaded. The max ID is calculated. The new record(s) is/are assigned max+1 (and max+2, max+3, etc.). They are uploaded one by one.

- ii. There is a bug in this process. If another device uploads during this sequence above, then it could overwrite data. Relax, though, as the window for this happening is less than one second. You have to sync two devices within one second of each other for there to be a problem.
 - iii. For projects not using autonumbering, a list of record IDs is also downloaded.
 - 1. If there is a conflict in an ID and a new record name, we treat the app record as *modified* and call this a *merge case*.
 - 2. For records without a conflict in IDs, the records are uploaded one by one.
- e. Modified records (technical)
- i. A report is compiled of all modified records. All of these records were downloaded before; otherwise, they would be new. A copy of the original data is stored in an unmodified format for this process.
 - ii. A copy of the current values on REDCap is also downloaded. Thus, we have three copies of data to adjudicate.
 - iii. The fields of the modified data for that record are compared against the current values.
 - 1. If equal, don't report.
 - 2. If unequal (i.e., if modified is not equal to current for a field):
 - a. If the unmodified is equal to current, then mark in *yellow* as having changed only on the app.
 - b. If the unmodified is unequal to the current, then mark in *pink/red* as having changed on the server *and* the app.
 - c. Give the user an option to choose on an adjudication table. Upload if the app field is chosen.
 - 3. If a record is *not present on the server* but present in the app and if we know that record was downloaded, then it was most likely deleted on the server. Therefore, we present the following options:
 - a. Delete the app record and don't upload.
 - b. Rename the app record to something else and upload the whole thing (not just the modified fields).
 - 4. If the record is a *merge case* (see above):
 - a. Rename the record and upload.
 - b. Merge the two records (current download and app version) field-by-field in an adjudication table. Upload accordingly.
 - c. Delete the app version altogether.

5. Uploads are done at a field-by-field level, not as a whole record. This prevents potential false overwrites of data that might be presently changing on the server.
 6. To prevent errors in large uploads, events are uploaded one-by-one in longitudinal projects. For classical projects, uploads occur one record at a time.
- f. Unmodified fields are not touched or uploaded.
 - g. All fields in uploaded records are marked unmodified.
 - h. How do I recover my data if syncing isn't working?
 - i. The Emergency Data Dump is designed for that purpose. It uploads a CSV (and any accompanying files) to your Mobile App File Archive available under the Mobile App section of your project. The CSV can then be imported into the Data Import Tool from your project.
 - ii. You can choose to send only New/Modified records in the Dump, or you can send everything. Be forewarned, though, that the Data Import Tool overwrites existing data. Sending everything can revert data values back to a previous form if they've changed on the REDCap server.
 - i. Next steps
 - i. The recommended next step is to download a new project configuration and perhaps a new set of data. (Either way, you are not allowed to keep your old data once you download a new configuration.) This ensures that you are kept current.
 - ii. Or you could keep your current setup and proceed regularly. Your old new and modified data has been marked as unmodified if the record has been uploaded.
 - iii. Or if your *admin* has enabled this button, you can just delete/clean your data and let your project configuration remain. This scenario is helpful for low-Internet-bandwidth situations.
 - j. Troubleshooting
 - i. What happens if the app hangs during this process? Try the '00' login process described above. Email redcapapp@vanderbilt.edu as well for help. Consider using the Emergency Data Dump.
 - ii. What happens if the app totally crashes during this process? You've probably run out of memory to some degree. The app only has a certain amount of memory from the operating systems. Try an Emergency Data Dump, which uses less memory.
 - iii. Sometimes, the *data dictionary/metadata* on the project changes. If a change throws data quality into jeopardy (by changing, say, a field label or a radio button text option), then the app throws an error. It will not upload the data until the data dictionary/metadata on the server is compatible.

1. That is, one must change back the metadata on the server first. Then sync the project. Then change forward the metadata. Then refresh the project's configuration and data.
2. The app will list which fields need attention. This is probably the most difficult process to organize on the human-end in the app.